



Proceedings of the Thirteenth
International Conference on
the Simulation and Synthesis
of Living Systems

edited by

Christoph Adami
David M. Bryson
Charles Ofria
Robert T. Pennock

Artificial Life 13

Artificial Life 13: Proceedings of the Thirteenth International
Conference on the Simulation and Synthesis of Living Systems

edited by

Christoph Adami, David M. Bryson,
Charles Ofria, Robert T. Pennock

An ISAL Book

The MIT Press
Cambridge, Massachusetts



Attribution-NonCommercial-NoDerivs 3.0 United States

You are free:



to Share — to copy, distribute and transmit the work

Under the following conditions:



Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).



Noncommercial — You may not use this work for commercial purposes.



No Derivative Works — You may not alter, transform, or build upon this work.

With the understanding that:

Waiver — Any of the above conditions can be **waived** if you get permission from the copyright holder.

Public Domain — Where the work or any of its elements is in the **public domain** under applicable law, that status is in no way affected by the license.

Other Rights — In no way are any of the following rights affected by the license:

- Your fair dealing or **fair use** rights, or other applicable copyright exceptions and limitations;
- The author's **moral** rights;
- Rights other persons may have either in the work itself or in how the work is used, such as **publicity** or privacy rights.

Notice — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to <http://creativecommons.org/licenses/by-nc-nd/3.0/us/>.

This is a human-readable summary of the [Legal Code \(the full license\)](http://creativecommons.org/licenses/by-nc-nd/3.0/us/legalcode) which can be found at <http://creativecommons.org/licenses/by-nc-nd/3.0/us/legalcode>.

Table of Contents

Preface	xv
---------------	----

Papers

Evolution in Action

The Role of Standing Genetic Variation in Adaptation of Digital Organisms to a New Environment	3
<i>Carlos J.R. Anderson</i>	
Towards the Recapitulation of Ancient History in the Laboratory: Combining Synthetic Biology with Experimental Evolution	11
<i>Betul Kaçar and Eric A. Gaucher</i>	
Digital Evolution Exhibits Surprising Robustness to Poor Design Decisions	19
<i>David M. Bryson and Charles Ofria</i>	
The Role of Deleterious Mutations in the Adaptation to a Novel Environment	27
<i>Arthur W. Covert III, Jared Carlson-Stevermer, Dakota Z. Derryberry and Claus O. Wilke</i>	
What Does Sex Have to do with it: Tracking the Fate of Deleterious Mutations in Sexual Populations	32
<i>Arthur W. Covert III, Lane Smith, Dakota Z. Derryberry and Claus O. Wilke</i>	
Heterochronous Neural Baldwinism	37
<i>Keith L. Downing</i>	
A Quantitative Measure of Non-Neutral Evolutionary Activity for Systems that Exhibit Intrinsic Fitness	45
<i>Alastair Droop and Simon Hickinbotham</i>	
Robustness and Evolvability of Cooperation	53
<i>Antoine Frenoy, François Taddei and Dusan Misevic</i>	

Evolution of Self-Replicating Cube Conglomerations in a Simulated 3D Environment	59
<i>Paul Grouchy and Hod Lipson</i>	
Open Ended Evolution of 3D Multicellular Development Controlled by Gene Regulatory Networks	67
<i>Michal Joachimczak and Borys Wrobel</i>	
Beyond Open-endedness: Quantifying Impressiveness	75
<i>Joel Lehman and Kenneth O. Stanley</i>	
Architectures for Self-Reproduction: Abstractions, Realisations and a Research Program	83
<i>Barry McMullin</i>	
Evolutionary Dynamics and Ecosystem Feedback in Two Dimensional Daisyworld	91
<i>Dharani Punithan and Bob McKay</i>	
Adaptation and Divergence during Experimental Evolution of Multicellular <i>Saccharomyces cerevisiae</i>	99
<i>Maria Rebolleda-Gomez, William Ratcliff and Michael Travisano</i>	
Comparing Distance-Based Phylogenetic Tree Construction Methods Using An Individual-Based Ecosystem Simulation, EcoSim	105
<i>Ryan Scott and Robin Gras</i>	
Evolved Modular Epistasis in Artificial Organisms	111
<i>Sergi Valverde, Ricard V. Solé and Santiago F. Elena</i>	
Evolutionary Potential is Maximized at Intermediate Diversity Levels	116
<i>Bess L. Walker and Charles Ofria</i>	
Is Evolution by Natural Selection the Algorithm of Biological Evolution?	121
<i>Richard A. Watson</i>	
Coevolving Parasites Improve Host Evolutionary Search on Structured Landscapes	129
<i>Hywel T.P. Williams</i>	
Sexual Selection, Resource Distribution, and Population Size in Synthetic Sympatric Speciation	137
<i>Mark Woehrer, Dean Hougen and Ingo Schlupp</i>	

Collective Dynamics

Contextual Geometric Structures: Modeling the Fundamental Components of Cultural Mehavior	147
<i>Bradly Alicea</i>	
Effects of Local Communication and Topology on Collective Movement Initiation	155
<i>Brent E. Eskridge</i>	
Evolving a Follower in the Presence of a Potential Leader	163
<i>Brent E. Eskridge</i>	
An Ecology-Based Evolutionary Algorithm to Evolve Solutions to Complex Problems	171
<i>Sherri Goings, Heather Goldsby, Betty H.C. Cheng and Charles Ofria</i>	
The Evolution of Temporal Polyethism	178
<i>Heather J. Goldsby, Neem Serra, Fred Dyer, Benjamin Kerr and Charles Ofria</i>	
Polarization and Belief Dynamics in the Black and White Communities: An Agent-Based Network Model from the Data	186
<i>Patrick Grim, Stephen B. Thomas, Steven Fisher, Christopher Reade, Daniel J. Singer, Mary A. Garza, Craig S. Fryer and Jamie Chatman</i>	
The Effects of Finite Populations and Selection on the Emergence of Signaling	194
<i>Kyle I. Harrington, A. Pinar Ozisik and Jordan B. Pollack</i>	
App Epidemics: Modelling the Effects of Publicity in a Mobile App Ecosystem	202
<i>Soo Ling Lim and Peter J. Bentley</i>	
An Ant-Based Computer Simulator	210
<i>Loizos Michael and Anastasios Yiannakides</i>	
Effects of Public Good Properties on the Evolution of Cooperation	218
<i>Dusan Misevic, Antoine Frenoy, David P. Parsons and François Taddei</i>	
Coevolutionary Dynamics between Roles and Social Sensitivity in an Extended Minority Game	226
<i>Keita Nishimoto, Ivan Tanev, Katsunori Shimohara, Reiji Suzuki and Takaya Arita</i>	

Characterizing Autonomy in the Web via Transfer Entropy	234
<i>Mizuki Oka and Takashi Ikegami</i>	
Evolution of Language through Messaging in Cooperative Tasks	243
<i>Aditya Rawal, Padmini Rajagopalan, Risto Miikkulainen and Kay Holekamp</i>	
odNEAT: An Algorithm for Distributed Online, Onboard Evolution of Robot Behaviours	251
<i>Fernando Silva, Paulo Urbano, Sancho Oliveira and Anders L. Christensen</i>	
Finding Optimal Random Boolean Networks for Reservoir Computing	259
<i>David Snyder, Alireza Goudarzi and Christof Teuscher</i>	
Representational Momentum May Explain Aspects of Vowel Shifts	267
<i>Samarth Swarup and Corrine McCarthy</i>	
Environment Classification in Multiagent Systems Inspired by the Adaptive Immune System	275
<i>Danesh Tarapore, Anders Lyhne Christensen, Pedro U. Lima and Jorge Carneiro</i>	
Evolutionary Transitions and Top-Down Causation	283
<i>Sara Imari Walker, Luis Cisneros and Paul C.W. Davies</i>	
The Role of Collective Working Memory in an Urban Pursuit Scenario	291
<i>Ransom K. Winder and James A. Reggia</i>	
 Behavior and Intelligence	
Second Order Learning and the Evolution of Mental Representation	301
<i>Solvi Arnold, Reiji Suzuki and Takaya Arita</i>	
On the Relationship Between Environmental and Mechanical Complexity in Evolved Robots	309
<i>Joshua E. Auerbach and Josh C. Bongard</i>	
Testing the Variability Selection Hypothesis: The Adoption of Social Learning in Increasingly Variable Environments	317
<i>James M. Borg and Alastair Channon</i>	

Evolutionary Design and Experimental Validation of a Flexible Caudal Fin for Robotic Fish	325
<i>Anthony J. Clark, Jared M. Moore, Jianxun Wang, Xiaobo Tan and Philip K. McKinley</i>	
Informational Drives for Sensor Evolution	333
<i>Sander G. van Dijk and Daniel Polani</i>	
Task Decomposition with Neuroevolution in Extended Predator-Prey Domain	341
<i>Ashish Jain, Anand Subramoney and Risto Miikkulainen</i>	
Brainless Bodies: Controlling the Development and Behavior of Multicellular Animats by Gene Regulation and Diffusive Signals	349
<i>Michal Joachimczak, Taras Kowaliw, Rene Doursat and Borys Wrobel</i>	
The Minimal Complexity of Adapting Agents Increases with Fitness	357
<i>Nikhil J. Joshi, Giulio Tononi and Christof Koch</i>	
On the Emergent Behaviors of a Robot Controlled by a Real-Time Evolving Neural Network	364
<i>Walter O. Krawec</i>	
Automatically Designing and Printing Objects with EvoFab 0.2	372
<i>Timothy Kuehn and John Rieffel</i>	
Rewarding Reactivity to Evolve Robust Controllers without Multiple Trials or Noise	379
<i>Joel Lehman, Sebastian Risi, David B. D'Ambrosio and Kenneth O. Stanley</i>	
Aracna: An Open-Source Quadruped Platform for Evolutionary Robotics	387
<i>Sara Lohmann, Jason Yosinski, Eric Gold, Jeff Clune, Jeremy Blum and Hod Lipson</i>	
Analysis of Evolved Agents Performing Referential Communication	393
<i>Santosh Manicka</i>	
Effects of Individual Differences on Knowledge and Wisdom of Society: A Social Modeling Approach	401
<i>Toshihiko Matsuka and Hidehito Honda</i>	
With a Little Help from Selection Pressures: Evolution of Memory in Robot Controllers	407
<i>Charles Ollion, Tony Pinville and Stéphane Doncieux</i>	

Evolved Neural Network Controllers for Physically Simulated Robots that Hunt with an Artificial Visual Sortex	415
<i>Michael E. Palmer and Andrew Chou</i>	

Evolution of Virtual Creature Foraging in a Physical Environment	423
<i>Marcin L. Pilat, Takashi Ito, Reiji Suzuki and Takaya Arita</i>	

Deformable Octahedron Burrowing Robot	431
<i>Juan Cristobal Zagal, Cristobal Armstrong and Shuguang Li</i>	

Synthetic Biology

Modeling Scalable Pattern Generation in DNA Reaction Networks	441
<i>Peter B. Allen, Xi Chen, Zack B. Simpson and Andrew D. Ellington</i>	

Energy-Based Artificial Chemistry Simulator	449
<i>Vincent Ducharme, Richard Egli and Claude Y. Legault</i>	

The Behavior-Based Hypercycle: From Parasitic Reaction to Symbiotic Behavior	457
<i>Tom Froese, Takashi Ikegami and Nathaniel Virgo</i>	

Checkpoint Orientated Cell-Cycle Simulation - Issues on Synchronised Situation	465
<i>Jonathan Pascalie, Valérie Lobjois, Herve Luga, Bernard Ducommun and Yves Duthen</i>	

Computational Tests of a Thermal Cycling Strategy to Isolate More Complex Functional Nucleic Acid Motifs from Random Sequence Pools by <i>in vitro</i> Selection	473
<i>Aaron Reba, Austin G. Meyer and Jeffrey E. Barrick</i>	

Design and Construction of a Prototype CMY (Cyan-Magenta-Yellow) Genetic Circuit as a Mutational Readout Device to Measure Evolutionary Stability Dynamics and Determine Design Principles for Robust Synthetic Systems	481
<i>Sean C. Sleight and Herbert M. Sauro</i>	

The Humanities and ALife

Using Pictures to Visualize the Complexity of Gene Regulatory Networks	491
<i>Sylvain Cussat-Blanc and Jordan Pollack</i>	

Finger-painting Fitness Landscapes: An Interactive Tool for Exploring Complex Evolutionary Dynamics.	499
<i>Luis Zaman, Charles Ofria and Richard E. Lenski</i>	

Extended Abstracts

Evolution in Action

The Influence of Genetic Operators and their Probabilities on the Lizards Behaviors within the Calangos Game	509
<i>Diego J. D. Almeida, Emanuel M. C. Tavares, Venyton N. L. Izidoro, Leandro N. De Castro, Angelo C. Loula and Charbel N. El-Hani</i>	
Evolution of Migratory-like Behavior in Avidians.	511
<i>Francis Bartlett, Fred C. Dyer and Robert T. Pennock</i>	
Evidence of Speciation in an Experimental Population of <i>E. coli</i> Following the Evolution of a Key Adaptation	513
<i>Zachary D. Blount and Richard E. Lenski</i>	
Cooperation and Antagonism in Information Exchange Between Two Species	515
<i>Andres C. Burgos and Daniel Polani</i>	
The Evolution of Modularity Under Changing Environments in Digital Organisms	517
<i>Rosangela Canino-Koning and Charles Ofria</i>	
Evolution of Aging and Rejuvenation in Bacteria	519
<i>Lin Chao, Camilla U. Rang and Annie Y. Peng</i>	
Specialization by <i>Burkholderia cenocepacia</i> Biofilm Ecotypes Limits Adaptation in a Planktonic Environment.	521
<i>Crystal N. Ellis, Rachel K. Staples and Vaughn S. Cooper</i>	
Experimental Evolution of an Artificial Bacterial Mutualism	523
<i>Kazufumi Hosoda, Akihiro Asao, Shingo Suzuki and Tetsuya Yomo</i>	

Darwinian Evolution of Translation-coupled RNA Self-replication System	525
<i>Norikazu Ichihashi, Kimihito Usui, Yasuaki Kazuta and Tetsuya Yomo</i>	
Network Representation of the Game of Life and Self-Organized Criticality	526
<i>Yoshihiko Kayama</i>	
Continuous <i>in vitro</i> Evolution of a Ribozyme Ligase: A Model Kit for The Evolution of a Biomolecule	528
<i>Michael P. Ledbetter, Tony W. Hwang, Gwendolyn M. Stovall and Andrew D. Ellington</i>	
Differences in the Concept of Fitness Between Artificial Evolution and Natural Selection	530
<i>Pawel Lichocki, Laurent Keller and Dario Floreano</i>	
Key Innovation in a Virus Catalyzes a Coevolutionary Arms Race	532
<i>Justin R. Meyer, Cesar Flores, Joshua S. Weitz and Richard E. Lenski</i>	
The Evolution of Allosteric Cooperativity in a Simple Artificial Life System	534
<i>Adam M. Novak, Anne E. Clark, Chris M. Deboever, Lillian E. Haynes, Singer Ma, Matt McDermott, John S. Wentworth and Eliot C. Bush</i>	
The Paradoxical Effects of Allelic Recombination on Fitness	536
<i>David P. Parsons, Carole Knibbe and Guillaume Beslon</i>	
Quantifying Frequency-Dependent Fitness Effects in Evolving Microbial Populations	538
<i>Noah Ribeck and Richard E. Lenski</i>	
Exploring the Concept of Open-Ended Evolution	540
<i>Tim Taylor</i>	
Size Does Matter: The Impact of Size on Hoarding Behaviour	542
<i>Olaf Witkowski and Nathanael Aubert</i>	
When Is Happy Hour: An Agent's Concept of Time	544
<i>Olaf Witkowski, Geoff Nitschke and Takashi Ikegami</i>	
Fitness Proportionate Sharing: a Different Perspective for Co-evolution of Diverse Population.	546
<i>Abrham Workineh and Abdollah Homaifar</i>	

Collective Dynamics

Implicit and Explicit Directional Information Transfer in Collective Motion	551
<i>Eliseo Ferrante, Ali Emre Turgut, Cristian Huepe, M. Birattari, M. Dorigo and T. Wenseleers</i>	
Diverse Behaviors in Swarm Robotics with Novelty Search	553
<i>Jorge Gomes, Paulo Urbano and Anders Lyhne Christensen</i>	
Computational Neuroecology of Communicated Somatic Markers	555
<i>Kyle I. Harrington, Megan M. Olsen and Hava T. Siegelmann</i>	
Evolutionary Chasing Between Cooperators and Defectors on the Spatial Prisoner's Dilemma	557
<i>Genki Ichinose, Masaya Saito and Shinsuke Suzuki</i>	
On Symbiotic Policy Search and Multi-level Selection	559
<i>Stephen Kelly, Peter Lichodziejewski and Malcolm I. Heywood</i>	
Limitations of Response Thresholds Models of Division of Labor	561
<i>Pawel Lichocki, Danesh Tarapore, Laurent Keller and Dario Floreano</i>	
A Synthetic Ecology Model for the Educational Game Calangos	563
<i>Angelo Loula, Leandro N. de Castro, Antônio L. Apolinário Jr, Pedro L. B. da Rocha and Charbel N. El-Hani</i>	
The Role of Memory in Stabilizing Swarms	565
<i>Jennifer M. Miller, Hao Luan, Louis F. Rossi and Chien-Chung Shen</i>	
A Bottom-Up Approach to the Evolution of Swarming	567
<i>Randal S. Olson, Christoph Adami, Fred C. Dyer and Arend Hintze</i>	
Can Simpson's Paradox Explain Co-operation in <i>Pseudomonas aeruginosa</i> Biofilms?	569
<i>Alexandra Penn, Tim C. R. Conibear, Richard A. Watson, Alexander R. Kraaijeveld and Jeremy S. Webb</i>	
"Take me to your leader!": Inferring Leadership in Animal Groups on the Move	571
<i>Nicolas Perony, Thomas O. Richardson, Marta B. Manser and Frank Schweitzer</i>	
Numerical Artificial Chemistries	573
<i>Juan Camilo Ramírez and James Marshall</i>	

Dynamic Phase Transition in a System of Self-propelled Particles	574
<i>Maksym Romenskyy and Vladimir Lobaskin</i>	
Evolutionary Swarm Chemistry in Three-Dimensions	576
<i>Hiroki Sayama</i>	
 Behavior and Intelligence	
Sustainable Population of Autonomous Foragers in a 3D Environment with Physics	581
<i>Nicolas Chaumont and Christoph Adami</i>	
Mapping the Collective Intelligence of the Artificial Life XIII Stakeholders	583
<i>Mark Dörr, Sif Schmidt-Petersen, Harold Fellermann, Lone Laursen and Steen Rasmussen</i>	
An Analysis of the <i>de novo</i> Evolution of a Complex Odometric Behavior	585
<i>Laura M. Grabowski, David M. Bryson, Fred C. Dyer, Robert T. Pennock and Charles Ofria</i>	
The Role of Local and Global Perspectives in the Dynamics of Opinion Convergence and Polarization	587
<i>Patrick Grim, Aaron Bramson, Daniel J. Singer, Steven Fisher, Carissa Flocken and William Berger</i>	
Evolution and Emergence of Sign Production and Interpretation	589
<i>Angelo Loula, Ricardo Gudwin and Joao Queiroz</i>	
Herd Behaviour Experimental Testing in Laboratory Artificial Stock Market Settings. Behavioural Foundations of Stylised Facts of Financial Returns	591
<i>Viktor Manahov and Robert Hudson</i>	
An Algorithm to Create Phenotype-Fitness Maps	593
<i>Jean-Baptiste Mouret and Jeff Clune</i>	
Multi-Robot, Multi-Patch Foraging with Maximum Sustainable Yield	595
<i>Zhao Song and Richard T. Vaughan</i>	
On-line, On-board Evolution of Reaction-Diffusion Control for Self-Adaptation	597
<i>Jürgen Stradner, Heiko Hamann, Payam Zahadat, Thomas Schmickl and Karl Crailsheim</i>	

Challenges for A-Life Approach to Artificial Cognition: in Search for Hierarchy of Cognitive Systems	599
<i>Borys Wrobel</i>	

Synthetic Biology

Protocellular Energetics and Autonomous Functions	603
<i>Anders Nikolaj Albertsen, Sara E. Maurer, Johnathan Cape, Harold Fellermann, James M. Boncella, Hans-Joachim Ziock, Steen Rasmussen and Pierre-Alain Monnard</i>	
Programming DNA-Based Reaction-Diffusion Circuits for Pattern Transformation	605
<i>Steven Chirieleison, Peter Allen, Andrew McIver, Alex Deiters, Andrew D. Ellington and Xi Chen</i>	
A Generic Graphical Interface For Multicellular Simulation	607
<i>Sylvain Cussat-Blanc, Jonathan Pascalie, Sylvain Tournois, Herve Luga and Yves Duthen</i>	
Compartmentalized Partnered Replication (CPR): A Generalizable Method for the Evolution of Biomolecules	609
<i>Jared W. Ellefson, Adam J. Meyer and Andrew D. Ellington</i>	
Generation and Screening of Genomic Libraries using <i>Mariner</i> Transposons and Cre/lox	611
<i>Peter J. Enyeart, Jeffrey E. Barrick, Scott P. Hunicke-Smith, Edward M. Marcotte and Andrew D. Ellington</i>	
An Artificial Multivesicular <i>in vitro</i> System to Emulate Multicellular Processes	613
<i>Maik Hadorn, Eva Boenzli, Martin M. Hanczyc, Steen Rasmussen and Peter Eggenberger Hotz</i>	
A Minimal Artificial Subcellular Matrix	615
<i>Maik Hadorn, Benny Gil, Carsten Svaneborg, Martin M. Hanczyc, Harold Fellermann, Rudolf Fuchslin, Peter Eggenberger Hotz, Casper Kunstmann-Olsen, Doren Lancet, John McCaskill, Pierre-Alain Monnard, Gunter von Kiedrowski and Steen Rasmussen</i>	
Towards Protocell Embedded Replication of Nucleic Acids	617
<i>Philipp M. G. Löffler, Rafal Wieczorek, Michael Wamberg, Mark Dörr, Pernille L. Pedersen, Carsten Svaneborg, Harold Fellermann, Joseph B. Edson, Jonathan L. Cape, Hans-Joachim Ziock, James M. Boncella, Steen Rasmussen and Pierre-Alain Monnard</i>	

Creating an Artificial Cell with Different Size Revealed the Effect of Compartment Volume on the Intracompartmental Multimeric Protein Synthesis	619
<i>Tomoaki Matsuura, Kazufumi Hosoda, Hiroaki Suzuki and Tetsuya Yomo</i>	
An Evolutionary-Genomics Approach for Elucidating and Improving Complex Microbial Phenotypes	621
<i>Jeremy Minty, Jihyang Park, Harris Wang, Lawrence Lai, Ted Zaroff III, Brian Johnson, Mark Burns, George Church and Xiaoxia Nina Lin</i>	
Statistical Analysis of Liposome Budding Dynamics Based on Free Energy Landscape	623
<i>Soichiro Tsuda, Hiroaki Suzuki and Tetsuya Yomo</i>	
The Origin of Life is a Spatially Localized Stochastic Transition	625
<i>Meng Wu and Paul G. Higgs</i>	
 The Humanities and ALife	
Automated Evolution of Interesting Images	629
<i>Joshua E. Auerbach</i>	
EndlessForms.com: Collaboratively Evolving 3D-Printable Objects Online	631
<i>Jeff Clune, Jason Yosinski, Eugene Doan and Hod Lipson</i>	
Germes, Genes, and Memes: Function and Fitness Dynamics on Information Networks	633
<i>Patrick Grim, Daniel J. Singer, Christopher Reade and Steven Fisher</i>	
Biology of Digital Organisms: How Language Constructs Reality	635
<i>Orly Kramash-Stettiner</i>	
The VIDA Art and Artificial Life Competition: Key Contributions to the Arts	637
<i>Nell Tenhaaf, Monica Bello Bugallo, Sonia Cillari, Jose Carlos Mariategui, Sally Jane Norman and Paul Vanouse</i>	
A Self-sustaining Visual Feedback Machine using Chaotic Neural Dynamics	639
<i>Alexander Woodward, Takashi Ikegami and Yuta Ogai</i>	
Author Index	641

Preface

These are the Proceedings of Artificial Life 13, the Thirteenth International Conference on the Simulation and Synthesis of Living Systems (<http://alife13.org/>), hosted by the BEACON Center for the Study of Evolution in Action (<http://beacon-center.org/>) at Michigan State University in East Lansing, Michigan, on July 19-22, 2012. The first Artificial Life Workshop was held at Los Alamos National Laboratory in September 1987, and the subsequent ALife workshops and conferences have been held biennially since 1990. These have been hosted previously in the U.S. eight times (Los Alamos 1987, Santa Fe 1990 & 1992, MIT 1994, UCLA 1998, Reed College 2000, Boston 2004), Japan once (Nara 1996), Australia once (Sydney 2002), England once (Southampton 2008), Denmark once (Odense 2010) and now again in the U.S. (East Lansing 2012).

Artificial Life, History, and the BEACON Center

This year marks the 25th anniversary of the Artificial Life conference series. The NSF-funded BEACON Center for the Study of Evolution in Action that is hosting this year's conference is perhaps the most visible outgrowth of these last 25 years. The Center, funded for 5 years with a budget of \$25 million and renewable for another 5 years, is a consortium of five universities led by Michigan State University. The other four members are North Carolina A&T State University, the University of Idaho, the University of Texas at Austin, and the University of Washington. The Center is focused on experimental and applied research on evolutionary dynamics, to understand and harness evolution as it happens ("in action") as opposed to the time-honored method of studying evolution by examining the past products of that process. The two main pillars of the experimental approach are the *E. coli* long-term evolutionary experiment (LTEE) begun by Dr. Richard Lenski about six months after the first ALife conference (Lenski, 2011), and digital life experiments with the Avida software. Avida, which saw the light of day 19 years ago, was first introduced to the scientific community during the 1994 ALife conference in Boston (Adami and Brown, 1994) and subsequently became a standard for research (Ofria and Wilke, 2004), as well as education (Pennock, 2007). The other leg of the BEACON Center is applied research using evolutionary computation methods, which was in part pioneered by the current director of the BEACON Center Dr. Erik Goodman with his advisor John Holland at the University of Michigan in 1972, forty years ago today. Clearly, many strands of ALife research have coalesced into this unique Center, and we are proud to be hosting ALife 13.

The ALife 13 Program

This year's conference was organized into five submission tracks: Evolution in Action, Behavior and Intelligence, Collective Dynamics, Synthetic Biology, and the Humanities and ALife. Of course, many submissions straddled tracks, or could have been assigned to two or more tracks, but these tracks also reflected well the keynote presentations, and allowed for a convenient organization into just a handful of categories. Dealing with over 200 submissions, assigning them to reviewers, scoring them and assigning them to sessions would not have been possible without the help of a dedicated group of track chairs, who deserve our gratitude and admiration for an often thankless, at times exasperating, but always time-consuming effort.

The *Evolution in Action* track was chaired by Drs. Claus Wilke and Santiago Elena, and covered contributions that throw light on evolutionary dynamics in general, including but not limited to concepts such as epistasis, pleiotropy, modularity, genotype-environment interactions, evolvability, robustness, speciation, evolution of sex, and the structure of fitness landscapes.

Behavior and Intelligence was the track chaired by Dr. Josh Bongard, with contributions that study animal and robot behavior, with the aim of understanding (from the bottom-up or the top-down) the algorithms behind animal and human decision making, as well as the conditions that give rise to complex sensory motor loops, intelligent behavior, cognition, and learning.

Collective Dynamics was chaired by Drs. Iain Couzin and Simon Garnier and was devoted to understanding groups of organisms, agents, or robots. Group behavior and dynamics encompasses (but is not limited to) the evolution and dynamics of cooperation, swarming, ecologies, food webs, crowd behavior, and biofilms.

The *Synthetic Biology* track, so prominent at the previous ALife conference was chaired by that conference's General Chair Dr. Steen Rasmussen. The track encompasses a wide variety of disciplines, ranging from the construction of new gene networks to re-engineering existing ones, over the construction of alternative genetics, to the synthesis of genomes and cells, to whole organisms.

Last but not least, the track devoted to the *Humanities and ALife*, spanning Art, Music, and Philosophy of Artificial Life, was ably chaired by Dr. Paula Gaetano Adi (Art and Music) and Dr. Patrick Grim (History and Philosophy). This broad track encompasses work in the humanities as it makes use of, interacts with or reflects upon the scientific activities and products of Artificial Life research.

Keynote Presentations

We were fortunate to have five superb keynote presenters for this year's meeting. We are grateful for the time they took to join us here, for their support of the conference series, and for their enthusiastic and entertaining talks.

Dr. Steven Benner of the Foundation for Applied Molecular Evolution in Florida is considered one of the pioneers of the field of Synthetic Biology, and delivered a keynote address entitled "*Artificial Life in Molecular Form*".

Dr. Oron Catts is the director of SymbioticA, the Centre of Excellence in Biological Arts, an artistic research laboratory housed within the School of Anatomy and Human Biology of the University of Western Australia. His keynote "*Are the Semi-Livings Art(ificial)? Shifting goalposts in relations to life*" entertained the audience on the night of the banquet dinner.

Dr. Benjamin Kerr (Department of Biology, University of Washington) is a pioneer in "Systems Evolutionary Biology", trying to understand evolution in the context of its environment, using experimental, computational, and theoretical methods. His keynote was titled "*From toxic bacteria to flammable plants: The evolution of altruism in structured communities*".

Dr. Radhika Nagpal (Department of Computer Science, Harvard University) focuses on engineering and understanding self-organizing systems, notably biologically-inspired multi-agent systems (collective algorithms), modular and swarm robotics, models of multicellular morphogenesis, and collective insect behavior. In keeping with this theme, her keynote was entitled "*Termite-like Robots and Robot-like Termites*".

Dr. Jack Szostak (Massachusetts General Hospital and Harvard Medical School) is the 2009 Nobel Laureate in Physiology or Medicine, and one of the key contributors in our quest to understand the origins of life via a constructive approach. Jack was also a keynote speaker at the ALife VI conference in 1998 conference (Adami et al., 1998), and we are lucky to be able to welcome him again. Because Jack is considerably more well known now, his keynote talk entitled "*The origin of life and the emergence of Darwinian evolution*" was held at MSU's Wharton Center and open to the public, to great acclaim.

Workshops & Tutorials

In addition to the peer-reviewed presentations, ALife 13 had five workshops and twelve tutorials. As in previous conferences, these were proposed and organized independently by individuals and groups from the ALife community. ALife workshops typically have a more open and flexible format set by the individual organizers and can cover a range of exploratory topics.

This year's workshops included:

Topic: Evolution of Physical Systems Workshop

Organizers: John Rieffel, Jean-Baptiste Mouret, Hod Lipson.

Topic: EvoNet2012: Evolving Networks, from Systems/Synthetic Biology to Computational Neuroscience

Organizers: Borys Wrobel, Maria Schilstra, Taras Kowaliw, Volker Steuber

Topic: Hard to Define Events Workshop

Organizer: Bradly Alicea and Laura Grabowski

Topic: Artificial Life in Industry

Organizer: Tom Barbalet

Topic: Teaching Artificial Life for Industry

Organizer: Tom Barbalet

Tutorials this year included:

Topic: Neuroevolution

Instructor: Risto Mikkulainen, University of Texas at Austin

Topic: Evolutionary Robotics

Instructor: Josh Bongard, University of Vermont

Topic: Evolutionary Game Theory

Instructor: Christoph Adami, Michigan State University

Topic: Genetic Algorithms

Instructor: Erik Goodman, Michigan State University

Topic: Genomic Analysis Tools

Instructor: Jeffery Barrick, University of Texas at Austin

Topic: Evolutionary Art

Instructor: Adam Brown, Michigan State University

Topic: Avida Artificial Life Platform

Instructors: Laura Grabowski, University of Texas - Pan American
David Bryson, Michigan State University

Topic: Avida-ED Digital Evolution Educational Software

Instructor: Robert T. Pennock, Michigan State University

Topic: Inner Workings of Avida

Instructor: David Bryson, Michigan State University

Topic: Fitness Landscapes and Epistasis

Instructors: Robert Heckendorn, University of Idaho
Charles Ofria, Michigan State University
Richard Lenski, Michigan State University
Ian Dworkin, Michigan State University

Topic: Quorum Sensing

Instructors: Chris Waters, Michigan State University
Brian Connelly, Michigan State University

Topic: SEEDS (Stochastic Ecological and Evolutionary Dynamics System)

Instructor: Brian Connelly, Michigan State University

The ALife 13 Process

The submission and review process was conducted and coordinated utilizing the distributed online tool EasyChair (<http://www.easychair.org/>). The quality of the conference depends, of course, upon the peer review of submissions by members of the Artificial Life community. We would like to thank the 149 researchers who served as submission reviewers for ALife 13. They are acknowledged by name below.

We, the ALife 13 Organizing Committee, sincerely hope you will find these Proceedings both useful and inspirational and that you will enjoy the conference.

Charles Ofria (General Chair)

Chris Adami (Program Chair)

Adam Brown (Art Advisory Committee Chair)

David M. Bryson (Technology & Proceedings Chair)

Erik Goodman (Engineering Advisory Committee Chair)

Connie James (Local Logistics Chair)

Taylor Kelsaw (Sponsorship Chair)

Richard Lenski (Scientific Advisory Committee Chair)

Phil McKinley (Workshops & Tutorials Chair)

Robert T. Pennock (Publicity & Proceedings Chair)

Danielle Whittaker (BEACON Coordination Chair)

July 2012, East Lansing, Michigan, U.S.A.

References

- Adami C, Belew RK, Kitano H, Taylor CE, editors (1998) Artificial Life VI: Proceedings of the Sixth International Conference on Artificial Life. Cambridge, MA: MIT Press.
- Adami C, Brown CT (1994) Evolutionary learning in the 2D artificial life system ‘Avida’. In: Brooks R, Maes P, editors. 6th International Conference on Artificial Life (ALife VI). Los Angeles, CA: MIT Press.
- Lenski RE (2011) Evolution in action: A 50,000-generation salute to Charles Darwin. *Microbe* 6: 30-33.
- Ofria C, Wilke CO (2004) Avida: a software platform for research in computational evolutionary biology. *Artificial Life* 10: 191-229.
- Pennock RT (2007) Learning Evolution and the Nature of Science using Evolutionary Computing and Artificial Life. *McGill Journal of Education* 42: 211-224.

Acknowledgments

We are grateful to many individuals and organizations who have contributed to the success of this conference. We owe a special thanks to the Department of Microbiology and Molecular Genetics of Michigan State University (MSU), which through its chair Dr. Walt Esselman supported scholarships for students to attend the conference, as well as Dr. Szostak's public lecture. Dr. Satish Udpa, the Dean of the MSU College of Engineering, also contributed to support the public outreach effort and was an enthusiastic supporter of the conference. We are also grateful to the MSU College of Social Sciences, the MSU Ecology, Evolutionary Biology, and Behavior Program (EEBB) through its director Dr. Kay Holekamp, the Department of Zoology and its chair Dr. Fred Dyer, and in particular to the BEACON Center for the Study of Evolution in Action, who provided funds that made it possible for numerous individuals to attend this conference through scholarships. We are especially grateful to the reviewers that accepted the task to participate in the peer review process, which no conference of this size can do without. While an evolutionary process relies materially on the generation of novelty via a random process, it is selection that guarantees that only the best variations make it to the next generation. Our reviewers provided this selection, and in essence created the document you now read. We thank:

Andy Adamatzky	Seth Bullock	Rene Doursat
Bradly Alicea	Chad Byers	Alastair Droop
Lee Altenberg	Ted Carmichael	Fred Dyer
Martyn Amos	Pedro Castillo	Bruce Edmonds
Carlos Anderson	Chris Chandler	Santiago Elena
Claes Andersson	Lauren Childs	Wesley Elsberry
Takaya Arita	Sung-Bae Cho	Brent Eskridge
Betül Arslan	Dominique Chu	Harold Fellermann
Joshua E Auerbach	Jeff Clune	Christoph Flamm
Carsten Baldauf	Brian D. Connelly	Rudolf M. Fölschlin
Wolfgang Banzhaf	Tim Cooper	Paula Gaetano-Adi
Jeffrey Barrick	Vaughn Cooper	Simon Garnier
Francis Bartlett	David Cornforth	Eric A. Gaucher
Anthony Beavers	Michael Cortez	Nicholas Geard
Peter Bentley	Iain Couzin	Philip Gerlee
Eleonora Bilotta	Art Covert	Carlos Gershenson
Zachary Blount	Thomas Dandekar	Mario Giacobini
Christian Blum	Kerstin Dautenhahn	Zann Gill
Johan Bollen	Peter Dittrich	Sherri Goings
Josh Bongard	Marco Dorigo	Heather Goldsby
Adam Brown	Alan Dorin	Erik Goodman

Laura Grabowski
Robin Gras
Virgil Griffith
Patrick Grim
Inman Harvey
Robert Heckendorn
Arend Hintze
Abdollah Homaifar
Paul Humphreys
Phil Husbands
Tim Hutton
Hiroyuki Iizuka
Takashi Ikegami
Christian Jacob
Sardanyés Josep
Gerald Joyce
George Kampis
Ben Kerr
David B. Knoester
Joel Lehman
Niles Lehman
Tom Lenaerts
Richard Lenski
Michael Levin
Xiaoxia Nina Lin
George Magoulas
Joanna Masel

Jerzy Maselko
Sarah Maurer
Philip Mckinley
Barry McMullin
Jj Merelo
Daniel Merkle
Justin Meyer
Jeremy Minty
Dusan Misevic
Andres Moya
Chrystopher L. Nehaniv
Marin Nilsson Jacobi
Jason Noble
Eckehard Olbrich
Naoaki Ono Elizabeth
Ostrowski
Bjørn Østman
Anuraag Pakanati
Michael Palmer
Simone Pigolotti
Daniel Polani
Erich Prem
Mikhail Prokopenko
Andres Ramirez
Steen Rasmussen
William C. Ratcliff
Thomas Ray

Noah Ribeck
John Rieffel
Kepa Ruiz-Mirazo
Matthew Rupp
Francisco C. Santos
Hiroki Sayama
Sean Sleight
Kenneth Stanley
Susan Stepney
Uwe Tangen
Tim Taylor
Michael Travisano
Soichiro Tsuda
Elio Tuci
Jon Umerez
Edgar Vallejo
Sergi Valverde
Richard Vaughan
L. Gwenn Volkert
Aaron Wagner
Bess Walker
Danielle Whittaker
Claus Wilke
Paul Williams
Michael Wiser
Lidia Yamamoto
Luis Zaman

Evolution in Action

Papers

The Role of Standing Genetic Variation in Adaptation of Digital Organisms to a New Environment

Carlos J. R. Anderson^{1,2,3}

¹Department of Zoology

²Ecology, Evolutionary Biology, and Behavior (EEBB) Program

³BEACON Center for the Study of Evolution in Action

Michigan State University, East Lansing, MI 48824

carlosja@msu.edu

Abstract

Evolutionary adaptation to a new environment depends on the availability of beneficial alleles. Beneficial alleles may appear as new mutations or may come from standing genetic variation—alleles already present in the population prior to the environmental change. Adaptation from standing genetic variation in sexually-reproducing populations is expected to be faster than from new mutations because beneficial alleles from standing genetic variation occur at a higher starting frequency and are immediately available. The distribution of fitness effects of alleles from standing genetic variation are expected to be different from that of new mutations because standing genetic variation has been ‘pre-tested’ by selection. Whether adaptation uses standing genetic variation or new mutations as a source of beneficial alleles is unknown. In this study, I conducted experimental evolution of digital organisms to determine the source of beneficial alleles during adaptation. I also tested the speed of adaptation and the fitness effect of alleles under these two sources of genetic variation. I found that the major source of beneficial alleles after an environmental change was standing genetic variation, but new mutations were necessary for long-term evolution. I also found that adaptation from standing genetic variation was faster than from new mutations, and the mean fitness effect of alleles from standing genetic variation were neutral, whereas new mutations were deleterious. Interestingly, I found that an important advantage of standing genetic variation was that recombination appeared to bring together beneficial combinations of alleles from standing genetic variation. These results support the hypothesis that adaptation occurs mostly from standing genetic variation and provide an additional advantage for such adaptation.

Introduction

When a population adapts to a new environment, beneficial alleles may appear as new mutations or come from standing genetic variation (Barrett and Schluter, 2008). Standing genetic variation refers to the presence of alternative alleles at each genetic locus in a population. Standing genetic variation may be maintained in a population for several reasons (Hartl and Clark, 1997); e.g., alleles with little or no effect on fitness may rise to moderate frequencies by random genetic drift. Standing genetic variation may be a major source of beneficial alleles in a new environment, with two

important implications for the dynamics of adaptation. First, adaptation from standing genetic variation should be faster than adaptation from new mutations because beneficial alleles would be immediately available and would be present at higher frequencies (Barrett and Schluter, 2008). Second, the distribution of fitness effects of alleles from standing genetic variation should be different than that of new mutations because standing genetic variation has been ‘pre-tested’ by surviving previous generations of selection against deleterious alleles (Barrett and Schluter, 2008).

Whether standing genetic variation is an important source of beneficial alleles for adaptation is unknown. Studies have employed three main approaches to answer this question (reviewed in Barrett and Schluter (2008)): analysis of the signature of selection, presence of the beneficial allele in the ancestral population, and phylogenetic analysis for inferring the history of alleles. These methods, however, are necessarily indirect and each has their unique set of problems. Of course, the “surest way to determine the source of beneficial alleles is to locate the genes themselves and establish their histories” (Barrett and Schluter, 2008). In this study, I used digital organisms to follow individual alleles through time as populations adapted to a new environment, and I determined whether beneficial alleles appeared as new mutations or came from standing genetic variation. I also tested whether adaptation from standing genetic variation was faster than from new mutations and whether the fitness effects of standing genetic variation were different from those of new mutations.

I conducted my experiments using Avida (Ofria and Wilke, 2004), an artificial life program designed to study questions in evolution, e.g., the complexity of epistasis (Lenski et al., 1999), the effect of mutational robustness on evolvability (Elena and Sanjuán, 2008), and the genetic architecture of sexual organisms (Misevic et al., 2006). Digital organisms in Avida consist of a sequence of computer instructions that encodes their ability to replicate and perform Boolean logic operations (or ‘tasks’). Variation in the efficiency of replication and in the ability to perform tasks arises via mutation and, in sexual organisms, recombination.

Organisms that are able to perform tasks are rewarded by allowing them to run more of their code per unit of time, effectively increasing their replication rate. Inheritance, variation, and differential reproduction in digital organisms allow them to evolve via natural selection and genetic drift. Thus, evolution in Avida is not simulated. The advantage of working with Avida is that one can run thousands of generations of experimental evolution in hours, perform replicate experiments with identical starting conditions, manipulate and analyze genomes easily, and record measurements like fitness with high accuracy.

Standing Genetic Variation in Digital Organisms

To generate a well-adapted, sexual population with standing genetic variation prior to the environmental change, I initialized an empty ‘world’ with an organism that could replicate but could not perform any tasks. I set the world size to 10,000 cells and the environment to reward for the default nine tasks (Lenski et al., 1999). I set the copy mutation rate to 0.1 mutations per genome per generation and, to ensure homologous recombination, I fixed the length of all genomes to 200 instructions and turned off insertion and deletion mutations. I let 50 such replicate populations evolve for 500,000 updates—a measurement of time in Avida—which was about 42,000 generations. I then picked a random population in which the consensus sequence could perform all nine tasks (35 out of the 50 could perform all nine tasks), and I took a random sample of 1,000 individuals from this population to serve as the ancestral population before the environmental change.

To measure the amount of standing genetic variation in the ancestral population, I measured the heterozygosity of each locus of the population. The heterozygosity of a locus is $H = 1 - \sum_{i=1}^k p_i^2$, where k is the number of alleles segregating at that locus and p_i is the frequency of the i th allele (Gillespie, 2004, p. 15). Here I adopted the convention that a locus is polymorphic (i.e., has standing genetic variation) if its most common allele has a frequency < 0.95 (Hartl and Clark, 1997, p. 53). A locus that had standing genetic variation would have a minimum heterozygosity of $1 - (0.95^2 + 0.05^2) = 0.095$. Because there are 26 possible

alleles (i.e., instructions) per locus in digital organisms, the maximum possible heterozygosity is approximately 0.9615.

I found substantial standing genetic variation in the ancestral population (Figure 1). Of 200 loci, 125 (62.5%) were polymorphic. The heterozygosity of each locus ranged from 0.0 to 0.8859, with a mean heterozygosity of 0.3781 (0.3334–0.4246, 95% bootstrap CI). For comparison, Stephens et al. (2001) found in humans that the heterozygosity of 313 genes ranged from 0.012 to 0.929, with a mean of 0.534. In natural populations of *E. coli*, Selander and Levin (1980) found that the heterozygosity of 20 enzyme-encoding genes ranged from 0.055 to 0.887, with a mean of 0.4718. My results demonstrate that the ancestral population exhibited levels of standing genetic variation consistent with that observed in biological populations. Furthermore, they support the claim that standing genetic variation is a ubiquitous property of evolving genetic systems (Gibson and Dworkin, 2004; Barrett and Schluter, 2008).

Source of Beneficial Alleles

Having established that the ancestral population harbored abundant standing genetic variation, I determined whether adaptation to a new environment relied on this genetic variation or on new mutations as a source of beneficial alleles. In this study, I examined beneficial alleles with fitness effects greater than 1%. With the ancestral population, I started 20 new replicate populations in a world of 1,000 cells and an environment that rewarded for 68 different tasks (the original nine tasks were not rewarded for). As a control, I also started another set of 20 replicate populations where every individual had an identical genotype (i.e., isogenic), set to the consensus sequence of the ancestral population. Although the consensus genotype did not actually exist in the ancestral population, its fitness was 1.0070 relative to the highest fit individual in the ancestral population (excluding those who could immediately perform tasks), and 1.0337 relative to the mean fitness of the ancestral population. Thus, the control population was not at a disadvantage compared to the ancestral population. All other configuration settings were identical to those used for the evolution of the ancestral population. Note that the populations that started with standing genetic variation were also allowed to get new mutations (the mutation rate was set to 0.1 mutations per genome

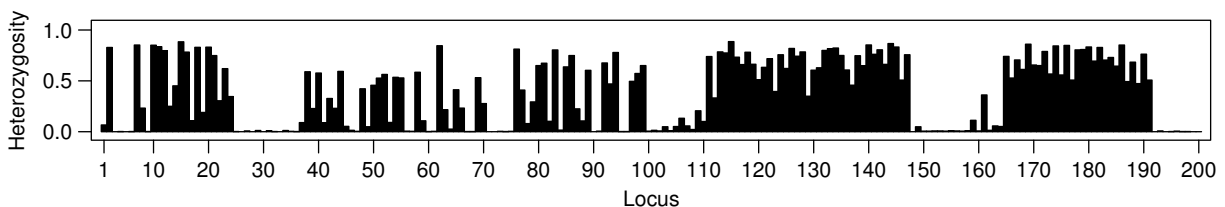


Figure 1: The heterozygosity of each locus of the population before the environmental change. Heterozygosities above 0.095 indicate the presence of standing genetic variation.

per generation). I let these replicate populations evolve for 10,000 updates (~ 850 generations), saving each population every 100 updates.

At the end of the runs, I found that the populations that started with standing genetic variation increased in mean fitness to 8.31 (7.74–8.87, 95% bootstrap CI) relative to the ancestral population in the new environment (i.e., the evolved populations were 8.31 times more fit in the new environment than the ancestral population). These populations were able to perform an average of 7.9 tasks, with a range of 5 to 10. The mean number of fixed, derived alleles—defined as having a frequency > 0.95 in the evolved population but < 0.95 in the ancestral population—was 56.25, ranging from 38 to 70. Figure 2 shows the history of two allele fixation events, one from standing genetic variation and the other from a new mutation, that occurred in the first replicate population. Of the 56.25 fixed, derived alleles, 47.8 (85%) existed as standing genetic variation in the ancestral population. In the control populations, mean fitness increased to 7.18 (6.62–7.76, 95% bootstrap CI) relative to the ancestral population. The control populations were able to perform an average of 6.7 tasks, with a range of 5 to 9. The mean number of fixed, derived alleles in the control populations was 5.15, ranging from 2 to 9. It was surprising that the populations that started with standing genetic variation fixed 10 times more alleles than the control populations, despite both sets of populations having similar final fitnesses and number of tasks performed.

The finding that 85% of fixed, derived alleles in the populations that started with the ancestral population existed as standing genetic variation may indicate that most beneficial alleles came from standing genetic variation. It is not clear, however, whether they were fixed by neutral genetic drift, natural selection, or genetic linkage and hitchhiking with beneficial alleles. For example, genetic hitchhiking in *Avida* can occur when alleles nearby a highly beneficial allele rise in frequency along with the beneficial allele. Hitchhiking occurs because the beneficial allele and nearby (i.e., genetically linked) alleles spread faster than recombination can break them apart. It is also not clear at what frequency the

derived alleles first became beneficial. Therefore, I developed a method to systematically measure the fitness of individual alleles through time and determine the frequency at which they became beneficial.

First, for each fixed, derived allele at the end of each run, I calculated both the allele's frequency and fitness effect every 100 updates, starting at the first update. To calculate the fitness effect of an allele at the current update, I first selected from the population the individual with the highest fitness who had the allele. I then created a clone of the individual and substituted the allele with an alternative allele drawn randomly from the standing genetic variation at that locus. I then calculated the fitness of the individual with the allele relative to the fitness of the individual without it. If this relative fitness was greater than 1.01, then the fitness effect of the allele ($> 1\%$) was beneficial at the current update. While testing this method, I found some cases where the fitness effect of the allele was considered beneficial only because the individual with the alternative allele had unusually low fitness. To reduce the frequency of such cases, I also required that the allele be beneficial for the individual with the second highest fitness. I stopped analyzing further updates as soon as I found the allele to be beneficial or if it became fixed.

In populations that started with standing genetic variation, I found that out of the mean 56.25 alleles that fixed, a mean of 31.9 became beneficial at some point in their history. I found that only 13.4% of these beneficial alleles became beneficial at a frequency < 0.05 (Figure 3, lower horizontal red line); the remaining 86.6% became beneficial at a frequency > 0.05 . Supposing standing genetic variation comprises alleles with frequencies > 0.05 , these results indicate that the majority of beneficial alleles came from standing genetic variation. In the control populations, I found that out of the mean 5.15 alleles that fixed, a mean of 5.1 became beneficial at some point in their history. I found that 77.3% of these beneficial alleles became beneficial at a frequency < 0.05 (Figure 3, upper horizontal red line); the remaining 22.7% became beneficial at a frequency > 0.05 . Therefore, in contrast to populations that started with standing genetic

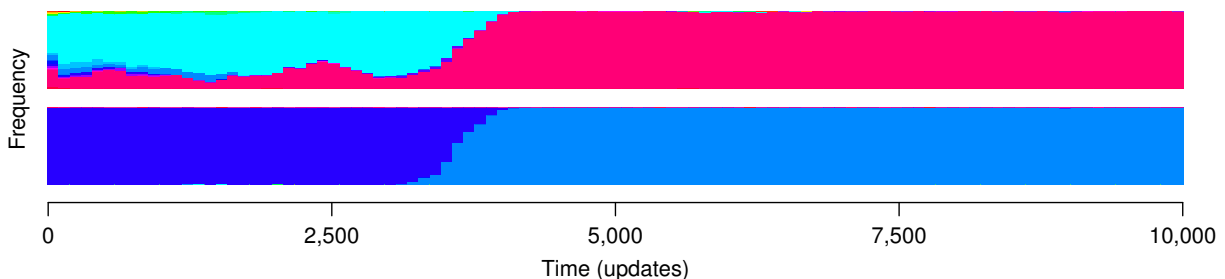


Figure 2: The frequencies of alleles through time for two loci in which an allele became beneficial and subsequently fixed. In the top plot, the beneficial allele came from standing genetic variation, and in the bottom plot, the beneficial allele appeared as a new mutation. Different alleles are represented by different colors. The y-axis in each plot ranges from 0.0 to 1.0.

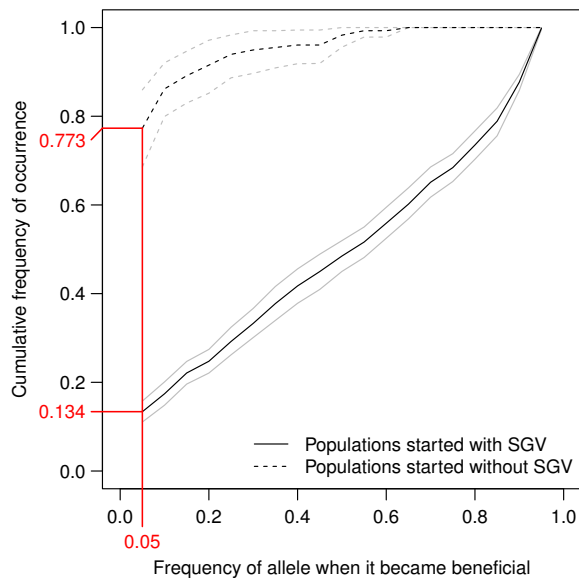


Figure 3: The cumulative frequency of fixed alleles that became beneficial at a specific frequency (0.05 bin size) for populations that started with standing genetic variation (solid lines) and for control, isogenic populations (dashed lines). The gray lines indicate the 95% bootstrap confidence interval around the mean of 20 replicate populations. The red vertical line indicates the frequency below which alleles were considered to appear as new mutations. The red horizontal lines indicate the proportions of alleles that came from new mutations for either type of population.

variation, the control, isogenic populations adapted mostly from new mutations, although almost a quarter of beneficial alleles came from standing genetic variation that arose as populations accumulated genetic polymorphism over time. Interestingly, the mean absolute (not percentage) number of new mutations per replicate for each treatment was about the same: 4.15 (3.40–4.85, 95% bootstrap CI) for populations started with standing genetic variation and 3.75 (3.3–4.2) for isogenic populations. This indicates that standing genetic variation did not inhibit new mutations from being selected.

One potential concern with the above method is that I identified beneficial alleles based on only two genotypes that had the allele, relative to two genotypes with alternative alleles. Yet the presumed beneficial alleles as well as the alternative alleles may not have the same fitness effect on other genetic backgrounds. Thus, I implemented a second method to identify beneficial alleles that considered more genotypes when measuring fitness effects. The key difference between this method and the previous is that in this method I selected all individuals who had the allele. Then, for each of these individuals I substituted the allele with an alternative allele drawn randomly from the standing genetic variation at that

locus. Finally, I calculated the mean fitness of all individuals with the allele relative to the mean fitness of all individuals with the allele replaced. If this relative fitness was greater than 1.01, then I considered the allele as beneficial. Using this method, I found that in populations that started with standing genetic variation, 11.5% of alleles became beneficial at a frequency < 0.05 ; the remaining 88.5% became beneficial at a frequency > 0.05 . In the isogenic populations, I found that 79.4% of alleles became beneficial at a frequency < 0.05 ; the remaining 20.6% became beneficial at a frequency > 0.05 . These results are very similar to those I found with the previous method, showing that the previous method was robust to the number of genotypes considered when identifying beneficial alleles.

Speed of Adaptation

Adaptation from standing genetic variation should be faster than adaptation from new mutations because beneficial alleles would be immediately available and would be present at higher frequencies (Barrett and Schluter, 2008). To test this prediction, I compared the speed of adaptation between populations that started with standing genetic variation and those that started with isogenic individuals. I re-evolved both types of populations at the additional mutation rates (U) of 0.01 and 0.0 (no new mutations) per genome per generation (the original populations were run at a mutation rate of 0.1). I added these new treatments because, given that the only source of mutations for the isogenic populations were new mutations, the mutation rate would be an important variable on the rate of adaptation. Population size would also be an important variable on the rate of adaptation, but I did not investigate its effects in this study.

I found that at the 0.1 mutation rate, the rate of adaptation for populations that started with standing genetic variation was significantly greater for most of the first four thousand updates than isogenic populations, then became less significantly so for the rest of the run (Figure 4A). At the 0.01 mutation rate, however, the rate of adaptation was significantly greater for the entire run (Figure 4B). Interestingly, at the 0.0 mutation rate, populations with standing genetic variation continued to adapt for several thousand updates, but, as expected, isogenic populations could not evolve (Figure 4C). These results clearly demonstrate that adaptation from standing genetic variation was faster than from new mutations. Yet new mutations were necessary for long-term evolution, as shown by the fact that adaptation from standing genetic variation without new mutations stopped after several thousand updates.

Fitness Effect of Random Alleles From Different Sources of Variation

The distribution of fitness effects of alleles from standing genetic variation should be different than that of new mutations because standing genetic variation has been ‘pre-tested’ by

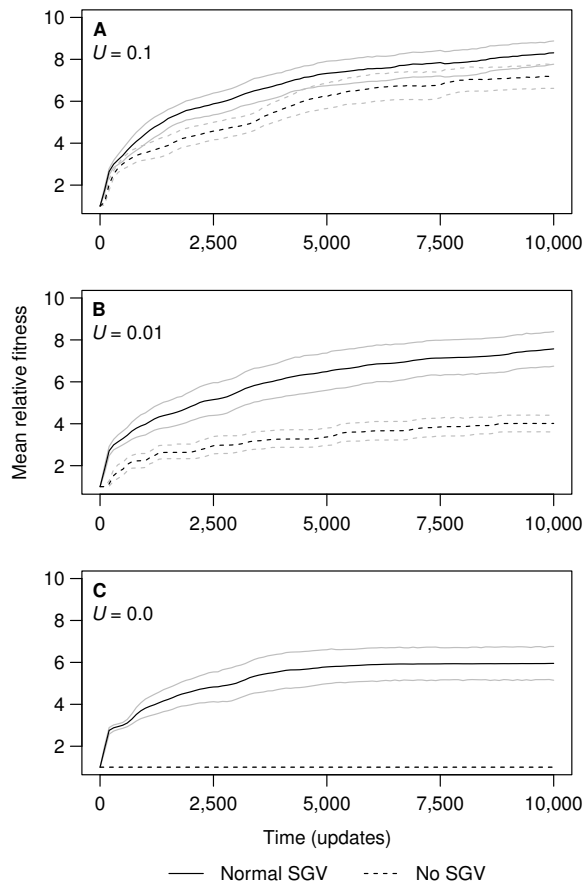


Figure 4: The mean fitnesses (relative to the ancestor) of populations evolved after an environmental change at (A) 0.1, (B) 0.01, and (C) 0.0 mutations per genome per generation (U). Populations evolved starting either with the ancestral population (solid line), which contained standing genetic variation (SGV) or with an isogenic population based on the consensus sequence of the ancestral population (dashed line). Gray lines represent the 95% bootstrap confidence intervals around the mean.

selection (Barrett and Schluter, 2008). To test this prediction, I generated the fitness effect distribution of alleles coming from either standing genetic variation or new mutations, measured in the new environment. First, I sampled 1,000 random (but viable) individuals from the ancestral population and mutated a single, random locus of each individual to an allele drawn randomly from the standing genetic variation (if there was any variation at that locus). I also sampled another set of 1,000 individuals from the ancestral population and mutated a single locus of each individual to an allele drawn randomly from all 25 possible alternative alleles. To prevent the possibility that these random mutations were more deleterious only because they disrupted fixed alleles, I ensured that the loci were drawn from the same pool of loci that had standing genetic variation. Finally, I measured the

Fitness effect	Source of mutation	
	SGV	Random
Lethal	0	58
Strongly deleterious	3	5
Mildly deleterious	186	345
Nearly neutral	729	520
Mildly beneficial	81	67
Strongly beneficial	1	5

Table 1: The number of single mutants (out of 1,000), categorized by the mutation's source and fitness effect (w): lethal ($w = 0$), strongly deleterious ($0 < w \leq 0.99$), mildly deleterious ($0.99 < w \leq 0.999$), neutral or nearly neutral ($0.999 < w \leq 1.001$), mildly beneficial ($1.001 < w \leq 1.01$), and strongly beneficial ($w > 1.01$).

fitness of these mutants relative to the original, unmutated individual.

I found that the mean fitness of mutants with mutations from standing genetic variation was 0.9994 (0.9969–1.0023, 95% bootstrap CI). The mean fitness of mutants with random mutations was 0.9496 (0.9326–0.9665, 95% bootstrap CI). Clearly, mutations from standing genetic variation did not have, on average, as strong deleterious effects as random mutations. To examine more closely the fitness effects of mutations from the two sources, I categorized each mutation based on the mutant's relative fitness (Table 1). Alleles from standing genetic variation were mostly neutral, whereas new mutations were more likely to be lethal or deleterious. Interestingly, new mutations were also more likely to be strongly beneficial than alleles from standing genetic variation, yet in the analysis where I determined the source of beneficial alleles, I found that most beneficial alleles came from standing genetic variation. This discrepancy may indicate that although alleles from standing genetic variation were not beneficial alone, combinations of these alleles brought together by recombination provided the benefits. The finding that alleles from standing genetic variation were less deleterious on average than random mutations support the hypothesis that standing genetic variation has been pre-tested by selection.

The above analysis was based on randomly generated mutants of the ancestral genotypes (i.e., at the beginning of the experiments), but it would also be interesting to know the fitness effect of beneficial alleles that actually fixed. This information was already calculated as part of determining the moment at which alleles became beneficial because it was used to determine whether alleles had achieved a fitness > 1.01 (using the first method). For populations that had evolved under standing genetic variation, the mean fitness of a genotype with a beneficial allele at the moment at which it became beneficial (relative to a genotype without the beneficial allele) was 1.54 (1.48–1.60, 95% bootstrap CI). For isogenic populations, this mean fitness was 1.47

(1.37–1.59, 95% bootstrap CI). Although the mean fitness effect of beneficial alleles for the standing genetic variation treatment was slightly higher than the isogenic treatment, they were not significantly different. The maximum relative fitness for a genotype with a beneficial allele for the standing genetic variation treatment (7.05) was higher than that for the isogenic treatment (4.50).

Discussion

I have shown that in populations of digital organisms adapting to a new environment, the major source of beneficial alleles was standing genetic variation, not new mutations. My findings are supported by selection experiments and observational studies of biological populations. Selection experiments have shown that adaptation can occur by changes in allele frequencies of standing genetic variation in the initial populations (e.g., Feder et al., 1997; Scarcelli and Kover, 2009; Teotónio et al., 2009). Observational studies of natural populations have found that alleles correlated with adaptive traits were also present in the ancestral population (e.g., Colosimo et al., 2005; Myles et al., 2005). In biological organisms, however, it is very difficult to measure the fitness effects of individual alleles, which is necessary to determine whether an allele fixed due to selection. Another problem, specific to studies of natural populations, is that the ancestral population is unavailable—the closest one can get is the extant population from which a subpopulation founded a new environment—and therefore it is often unknown whether a beneficial allele existed as standing genetic variation. The use of digital organisms allowed me to track individual alleles through time and determine the frequency at which they became beneficial.

When alleles from standing genetic variation became beneficial, their starting frequency ranged from the minimum of 5% to the maximum of 95% (Figure 3). In experimental studies of biological organisms, high starting frequencies (> 50%) are not uncommon (e.g., Feder et al., 1997; Scarcelli and Kover, 2009). In natural populations, however, starting frequencies have tended to be much smaller, such as in the study by Colosimo et al. (2005), where the starting frequency of an adaptive allele was between 0.2% and 3.8% in the ancestral population. One possible reason for this discrepancy is that natural populations may be under stronger selective pressures than experimental populations (Ellegren and Sheldon, 2008), so the fitness effects of alleles in natural populations tend to be more deleterious and therefore maintained at low frequencies. Of course, allele frequency data for adaptive alleles in natural populations is scarce, so more research in natural populations should determine the frequencies at which alleles from standing genetic variation become beneficial.

Adaptation should be faster if most beneficial alleles came from standing genetic variation than if they came from new mutations (Barrett and Schluter, 2008). I found this to be

the case in digital organisms if the mutation rate was low enough (Figure 4). In fact, when no new mutations were allowed, adaptation by standing genetic variation continued for several hundred generations, whereas no adaptation occurred in isogenic populations. Still, the importance of new mutations for long-term evolution was shown by the fact that adaptation stopped eventually when no new mutations were allowed. Although there are no empirical studies testing the speeds of adaptation, where beneficial alleles may come from either standing genetic variation or new mutations, my results are supported theoretically (Hermisson and Pennings, 2005). There are two reasons that adaptation from standing genetic variation should be faster than adaptation from new mutations: beneficial alleles are both readily available and present at higher frequencies than alleles from new mutations (Barrett and Schluter, 2008), which must overcome drift because they start at lower frequencies. Future experiments should be able to quantify the relative contribution of these two causes.

Although not examined in detail in this study, the population size and mutation rate can affect the relative contributions of standing genetic variation and new mutations during adaptation. For example, a sudden decrease in population size (i.e., a bottleneck) will reduce both the amount of standing genetic variation and the number of new mutations that appear each generation. In this case, standing genetic variation will still have an advantage over new mutations—especially for alleles of weak fitness effect—because weak effect alleles introduced by new mutations are easily lost due to genetic drift (Hermisson and Pennings, 2005). For large effect alleles, standing genetic variation will have a reduced advantage because large effect alleles are less likely to be lost even if they are introduced as new mutations (Hermisson and Pennings, 2005). In my experiments, mutations that allowed organisms to perform new tasks were of large effect (the default configuration in *Avida*), but future studies should experiment with weaker beneficial alleles. In a large population or high mutation rate, new mutations would become more important because large-effect mutations would appear more frequently.

Because alleles from standing genetic variation have had a potentially long history in an evolving population, their fitness effects in a new environment have been predicted to be less deleterious than random mutations (Barrett and Schluter, 2008). On average, I found that standing genetic variation was effectively neutral (fitness effect of 0.0006), whereas random mutations were strongly deleterious (fitness effect of 0.0504). Alleles from standing genetic variation can therefore linger in a population, increasing the chance for them to become beneficial after an environmental or genetic change. Random mutations, on the other hand, are on average deleterious and are thus more easily eliminated by selection. In biological populations, the mean fitness effect of random mutations was found to be 0.48 in RNA viruses

(Sanjuán et al., 2004), 0.12 in *C. elegans* (Vassilieva et al., 2000), and 0.22 in yeast (Zeyl and DeVisser, 2001). There are no measurements of the fitness effects of alleles from standing genetic variation in a biological population in a new environment.

For strongly beneficial mutations (i.e., fitness effect > 1%), I found that random mutations were more likely to be beneficial than alleles from standing genetic variation in the new environment (Table 1). It may thus seem counter-intuitive that most beneficial alleles during adaptation came from standing genetic variation. I hypothesize that it was the combination of many alleles from standing genetic variation that provided the benefits, and together these epistatically related alleles rose to fixation. Adaptation that requires many alleles working together is known as ‘polygenic adaptation’ (Pritchard and Di Rienzo, 2010), although fixation of alleles is not always necessary. In fact, Pritchard and Di Rienzo (2010) hypothesize that if adaptation occurs from standing genetic variation, polygenic adaptation is likely.

In summary, this study has shown the importance of standing genetic variation in populations of digital organisms adapting to a new environment. That is, (1) most beneficial alleles came from standing genetic variation rather than from new mutations, (2) populations that started with standing genetic variation adapted faster than populations that started with identical genotypes, and (3) the fitness effects of alleles from standing genetic variation were less harmful than new mutations. Because digital organisms evolve by the same processes of natural selection and genetic drift that biological populations also experience, I suspect that the above points are also true for biological populations. A hypothesis that arose from this study was that standing genetic variation together with recombination may give rise to combinations of alleles that together are beneficial. Future work should test whether this additional advantage is true, thereby highlighting the importance of sexual recombination and standing genetic variation in evolving populations.

Acknowledgments

I thank B. L. Williams for his guidance during the development of this study, C. Ofria and L. Harmon for their review of the manuscript, the members of the MSU Digital Evolution Laboratory for discussion, and four anonymous reviewers that helped improve the quality of this study. This material is based in part upon work supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

References

- Barrett, R. D. H. and Schluter, D. (2008). Adaptation from standing genetic variation. *Trends Ecol. Evol.*, 23:38–44.
- Colosimo, P. F., Hosemann, K. E., Balabhadra, S., Villarreal Jr., G., Dickson, M., Grimwood, J., Schmutz, J., Myers, R. M., Schluter, D., and Kingsley, D. M. (2005). Widespread parallel evolution in sticklebacks by repeated fixation of ectodysplasin alleles. *Science*, 307:1928–1933.
- Elena, S. F. and Sanjuán, R. (2008). The effect of genetic robustness on evolvability in digital organisms. *BMC Evol. Biol.*, 8:284.
- Ellegren, H. and Sheldon, B. C. (2008). Genetic basis of fitness differences in natural populations. *Nature*, 452:169–175.
- Feder, J. L., Roethele, J. B., Wlazlo, B., and Berlocher, S. H. (1997). Selective maintenance of allozygote differences among sympatric host races of the apple maggot fly. *Proc. Natl. Acad. Sci. USA*, 94:11417–11421.
- Gibson, G. and Dworkin, I. (2004). Uncovering cryptic genetic variation. *Nature Rev. Genet.*, 4:681–690.
- Gillespie, J. H. (2004). *Population genetics: a concise guide*. Johns Hopkins University Press, Baltimore, MD.
- Hartl, D. L. and Clark, A. G. (1997). *Principles of population genetics*. Sinauer Associates, Inc., Sunderland, MA.
- Hermisson, J. and Pennings, P. S. (2005). Soft sweeps: molecular population genetics of adaptation from standing genetic variation. *Genetics*, 169:2335–2352.
- Lenski, R. E., Ofria, C., Collier, T. C., and Adami, C. (1999). Genome complexity, robustness, and genetic interactions in digital organisms. *Nature*, 400:661–664.
- Misevic, D., Ofria, C., and Lenski, R. E. (2006). Sexual reproduction reshapes the genetic architecture of digital organisms. *Proc. R. Soc. B*, 273:457–464.
- Myles, S., Bouzekri, N., Haverfield, E., Cherkaoui, M., Dugoujon, J.-M., and Ward, R. (2005). Genetic evidence in support of a shared eurasian-north african dairying origin. *Hum. Genet.*, 117:34–42.
- Ofria, C. and Wilke, C. O. (2004). Avida: a software platform for research in computational evolutionary biology. *Artif. Life*, 10:191–229.
- Pritchard, J. K. and Di Rienzo, A. (2010). Adaptation – not by sweeps alone. *Nature Rev. Genet.*, 11:665–667.

- Sanjuán, R., Moya, A., and Elena, S. F. (2004). The distribution of fitness effects caused by single-nucleotide substitutions in an rna virus. *Proc. Natl. Acad. Sci. USA*, 101:8396–8401.
- Scarcelli, N. and Kover, P. X. (2009). Standing genetic variation in *FRIGIDA* mediates experimental evolution of flowering time in *Arabidopsis*. *Mol. Ecol.*, 18:2039–2049.
- Selander, R. K. and Levin, B. R. (1980). Genetic diversity and structure in *Escherichia coli* populations. *Science*, 210:545–547.
- Stephens, J. C. et al. (2001). Haplotype variation and linkage disequilibrium in 313 human genes. *Science*, 293:489–493.
- Teotónio, H., Chelo, I. M., Bradić, M., Rose, M. R., and Long, A. D. (2009). Experimental evolution reveals natural selection on standing genetic variation. *Nat. Genet.*, 41:251–257.
- Vassilieva, L. L., Hook, A. M., and Lynch, M. (2000). The fitness effects of spontaneous mutations in *Caenorhabditis elegans*. *Evolution*, 54:1234–1246.
- Zeyl, C. and DeVisser, J. A. G. M. (2001). Estimates of the rate and distribution of fitness effects of spontaneous mutation in *Saccharomyces cerevisiae*. *Genetics*, 157:53–61.

Towards the Recapitulation of Ancient History in the Laboratory: Combining Synthetic Biology with Experimental Evolution

Betül Kaçar^{1,2} and Eric Gaucher^{2,3}

¹NASA Astrobiology Institute, USA

²School of Biology, Georgia Institute of Technology, Atlanta, GA, 30322, USA

³School of Chemistry, Parker H. Petit Institute of Bioengineering and Biosciences,
Georgia Institute of Technology, Atlanta, GA, 30332, USA
betul@gatech.edu

Abstract

One way to understand the role history plays on evolutionary trajectories is by giving ancient life a second opportunity to evolve. Our ability to empirically perform such an experiment, however, is limited by current experimental designs. Combining ancestral sequence reconstruction with synthetic biology allows us to resurrect the past within a modern context and has expanded our understanding of protein functionality within a historical context. Experimental evolution, on the other hand, provides us with the ability to study evolution in action, under controlled conditions in the laboratory. Here we describe a novel experimental setup that integrates two disparate fields - ancestral sequence reconstruction and experimental evolution. This allows us to rewind and replay the evolutionary history of ancient biomolecules in the laboratory. We anticipate that our combination will provide a deeper understanding of the underlying roles that contingency and determinism play in shaping evolutionary processes.

Introduction

Living organisms are the product of their histories. Evolutionary biology is therefore an inherently historical science yet many details of this history are unobtainable: the fossil record is incomplete; ancestral genomic sequence information has been over-written via mutations; natural evolution occurs on long time scales; and the connections between genotype and phenotype are often intractable. Understanding these details is particularly difficult when one considers the potential role that chance plays in evolutionary outcomes. Along these lines, Stephen Jay Gould once remarked:

[H]istory includes too much contingency, or shaping of present results by long chains of unpredictable antecedent states, rather than immediate determination by timeless laws of nature... (Gould 1994).

Gould's remark suggests that there are too many solutions for life to be repeatable. Such a suggestion implies that historical contingency is a fundamental determinant of evolutionary outcomes. Others, such as Simon Conway Morris, have argued that evolution is actually highly constrained, with many available pathways to only a relatively few destinations (Morris 2003). Advances in the field of experimental evolution and whole-genome sequencing now make it possible to empirically examine the role of historical contingency in evolution at both the organismal (Wichman et al. 2000; Counago et al. 2006; Blount et al. 2008; Pena et al. 2010; Meyer et al. 2012) and molecular levels (Weinreich et al. 2006; Poelwijk et al. 2007; Pennisi 2011; Salverda et al. 2011).

While various experimental evolution approaches have made much progress in dissecting the role of history in evolution by directly observing evolution in action, less is known about the direct relationship between genotypes (modern or ancient) and their effect on shaping an organism's evolutionary trajectory. Here we propose a novel synthesis of synthetic biology and experimental evolution that will further our understanding by combining molecular and systems evolution and provide an unprecedented means of addressing how contingency and deterministic forces interact to guide evolutionary trajectories.

Rebuilding History and Creating Novelty with Synthetic Biology

Synthetic biologists assemble DNA to construct novel genes, metabolic pathways and even organisms (Benner and Sismour 2005; Endy 2005; Gibson et al. 2010). These manipulations provide us with a level of control

that natural systems cannot provide, and this level of control minimizes unknown variables/parameters that effect particular systems. A powerful and increasingly useful synthetic biological approach is the computational reconstruction of ancient sequences of biomolecules using Ancestral Sequence Reconstruction (ASR), an approach sometimes referred to as paleogenetics. Initially proposed by Pauling and Zuckerkandl (Pauling and Zuckerkandl 1963), ASR merges history with natural selection (Stackhouse et al. 1990). ASR involves the alignment of DNA or protein sequences, followed by the construction of a phylogenetic tree that is then used to infer sequences of ancestral genes at interior nodes of a tree using likelihood and/or Bayesian statistics (Gaucher 2007). Recent advances in DNA synthesis now permit us to resurrect these ancient sequences in the laboratory and recombinantly express the ancient genes using modern organisms *in vivo* or reconstituted *in vitro* translation systems. Through a bottom-up approach we can engineer novel artificial systems that can be manipulated to better understand nature. The growing list of resurrected biomolecules now includes hormone receptors (Thornton et al. 2003), alcohol dehydrogenases (Thomson et al. 2005), elongation factors (Gaucher et al. 2008), thioredoxins (Perez-Jimenez et al. 2011), among others (Benner et al. 2007) and most recently, complex molecular machines (Finnigan et al. 2012).

As ASR follows a bottom-up approach and utilizes modern sequences to infer the past states of biomolecules, experimental evolution pursues a top-down approach that involves the real-time examination of the evolution of microbial model systems (Figure 1). (in the present context, top-down refers to complex cellular systems and/or to whole organisms). Experimental evolution has been used to address important questions in evolutionary biology (Elena and Lenski 2003). The experimental evolution approach is particularly powerful because of the high level of control it permits, the tractability of its microbial participants, and the capacity to create and maintain a viable frozen fossil record of the evolving populations that may then be used for highly detailed studies to address a variety of questions in evolutionary biology.

Here we introduce for the first time a novel system in which ASR is combined with experimental evolution, we term *paleo-experimental evolution*. In this approach, ASR is used to reconstruct an ancestral gene/protein. The synthetic ancestral gene is then used to precisely replace the endogenous form of the gene from a modern

organism at the exact same chromosomal location. In some instances, we expect this replacement to cause the modern organism to be maladapted because the ancient gene/protein is not functionally equivalent to its (modern) descendent homolog. This synthetic recombinant organism is then experimentally evolved in the laboratory, and the subsequent adaptations are monitored using fitness measurements and whole-genome sequencing.

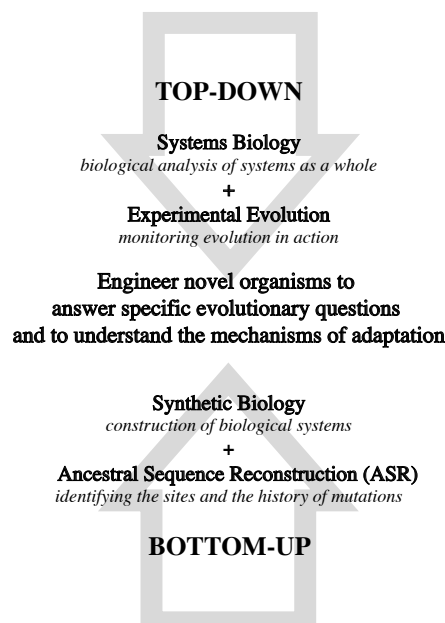


Figure 1: Artificial biology meets nature. In a novel paleo-experimental evolution system, descriptive evolutionary biology (top-down) meets applied, engineered synthetic biology (bottom-up) to further our understanding of evolutionary mechanisms

A paleo-experimental evolution setup also allows us to rewind and replay the molecular tape of life (or more precisely, one biomolecular component of life) to understand the role of chance and determinism in evolution, albeit in a laboratory setting. If evolutionary outcomes are deterministic, placing ancestral proteins within a modern context may result in the convergence of the ancient sequence towards the sequence of its modern counterpart. Alternatively, were historical contingency to be a major determinant of organismal evolution, there should be a number of available fitness peaks that may or may not be equally optimal and accessible via multiple trajectories. A major challenge, however, lies in our ability to develop a system that permits adaptation to occur along both deterministic and contingent paths if given equal *a priori* opportunity. Of course it is difficult to conceive of such an ideal system. However, we should be able to manage some aspects of

such a system. For instance, if we choose to evolve an ancient enzyme that binds to only a single substrate and converts that substrate into a product subsequently used downstream in a metabolic pathway, then we are limited in the trajectories that the ancient enzyme can adapt (the enzyme can evolve or the substrate can change). On the other hand, if we choose to evolve an ancient enzyme that has numerous substrates and binds many ancillary protein partners, then we can expect such a system to evolve more contingently than the previous scenario because there is greater opportunity for compensatory co-evolution to overcome the low fitness of the ancestral protein when placed in a modern context. Again, the enzyme may evolve or the enzyme's substrate may change. Unique to this scenario, however, is the potential for interacting protein partners to accumulate mutations that restore interactions otherwise diminished by the ancestral protein.

Ancient Hubs in Modern Times

A paleo-experimental evolution system that combines synthetic and evolutionary biology requires a deep understanding of the interactions of cellular components, biological networks, and gene regulation and expression. These components are shaped by the interplay between genotype and phenotype – the major determinants of natural selection.

What is fascinating in this complex picture is the harmonious *dialect*, a manner of language defined by intermolecular interactions within the context of the cell and that has the ability to respond and adapt to varying environments (Dennett 1995). Such a dialect can be a fine-tuned product of millions of years of evolutionary history both between and within the components of a cellular system. This very point challenges our ability to design new biological partners: fundamentally we are restricted by an organism's past.

Interchanging a modern protein in a cell with its homologous counterpart from another species can provide insight into the evolutionary paths and constraints that shape the evolution of homologous proteins. However, this intriguing experiment can fail to capture that the two homologs do not have a direct line of descent that *connects* them in progressive, linear time. Meaning, the evolutionary path that connects the two homologs requires that we travel *back* in time from one descendent to the common ancestor and then *forward* in time to the other descendent. As such, the homologs share a common ancestor but the descendents of that common ancestor traversed two separate (and possibly non-interchangeable) paths of adaptation and random fixation. This raises the possibility that the homologs are not 'functionally equivalent' (Figure 2A).

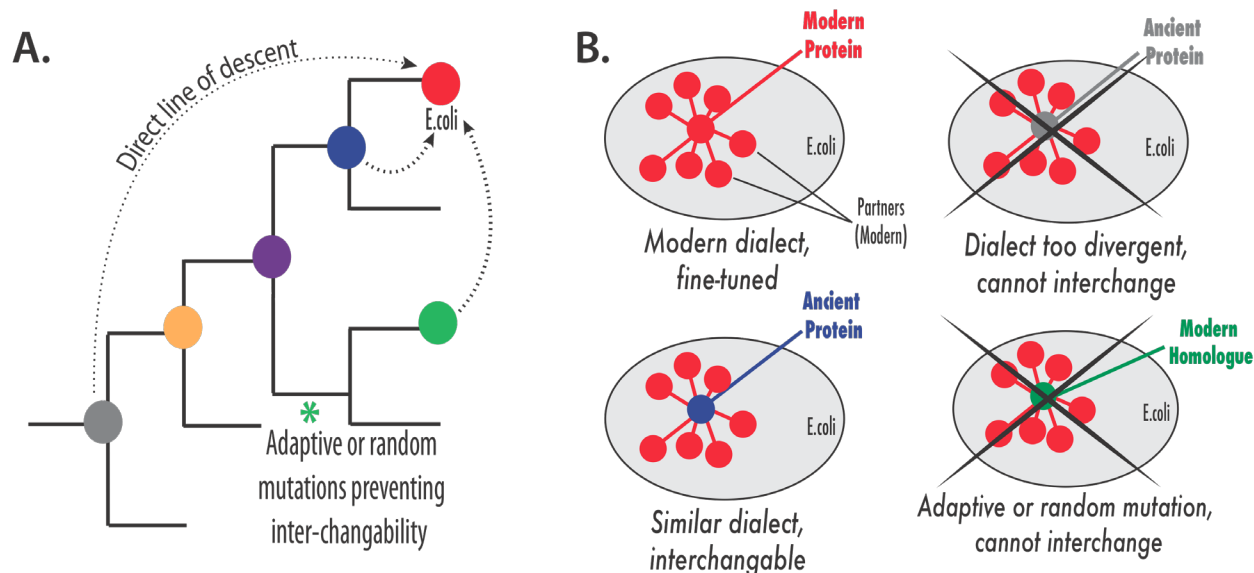


Figure 2. (A) Paleo-experimental evolution consists of resurrecting an ancient gene, removing the modern form of the gene from an extant organism, and then inserting the ancestral form into the extant organism. For instance, the ancient gene from the gray node on the phylogeny can be resurrected and then inserted into the *E. coli* genome (red node) at the precise chromosomal location that the extant gene was knocked out. This synthetic/engineered organism is then evolved in the laboratory. Our approach contrasts to other approaches that are only able to use modern genes from an organism to replace its ortholog in a different extant organism (say,

inserting the gene from the extant organism at the green node into the *E. coli*, red, organism). Such an approach can be limiting if adaptive or neutral mutations that prevent interoperability occurred along particular branches that connect the green and red nodes. (B) Protein interaction network containing modern and ancient hubs. Consider a particular hub protein that interacts with seven ancillary partners in *E. coli* (upper left). These interactions are fine-tuned over the course of evolution. Replacing the modern hub of the network with a recent ancestor of *E. coli* (blue) may permit the interaction network to still function, likely in a diminished capacity. Replacing the modern hub of the network with an ancient ancestor of *E. coli* (gray) may prevent the ancillary proteins from interacting with the hub altogether. Similarly, replacing the modern hub of the network with a divergent modern counterpart may prevent the interaction network from functioning despite that the same network exists in both modern organisms.

One manner in which homologs can become functionally nonequivalent is if a protein is part of a highly integrated molecular network in which the protein interacts with numerous ancillary partners. Sufficient co-adaptation or compensatory co-evolution amongst the protein and its ancillary partners along any phylogenetic lineage may prevent that particular protein from binding its necessary ancillary partners when interchanged in a different species (Figure 2B).

Replacing network partners with their ancestors would permit us to rewire a network within the historical context from which the mutational differences between the modern and ancestral proteins share a direct connection in evolutionary time. In a scenario where the hub (center) and nodes (terminal) of an interaction network have adapted to a particular lineage-specific dialect, replacing a component of such a network with its ancient counterpart may be analogous to resurrecting an old dialect that can be understood by its descendant speakers. As expected, the ability of the rest of the network to communicate (function) with this component from an ancient dialect would be limited by the manner in which the network changed between the ancestor and its modern form. The question of interest to us is whether the different components are capable of communicating or whether they will fail to communicate and thus be functionless – whoa is the Tower of Babel. We anticipate that ancestral components will in fact be able to communicate in the hub better than modern components from different species as long as the ancestor lies along the evolutionary path that directly connects the two modern proteins.

Despite our optimism, we suspect that the ancestral component will trigger a stress or strain on the modern network since the ancestral protein comes from an ancient dialect. If so, this creates an ideal scenario to watch the ancient component adapt within a modern network. Four possible scenarios may arise from such a system:

- 1) The ancient protein repeatedly adapts to the modern network in a manner identical or different to how its modern counterpart evolved (determinism).
- 2) The ancient protein adapts to the modern network in a manner different than how its descendent did (contingency).
- 3) The modern network adapts to the ancient protein in a manner identical to the ancient network - thus resurrecting the ancient network.
- 4) The modern network adapts to the ancient protein in a manner never evolved before in nature – thus creating an entirely new dialect.

The ability to differentiate these scenarios will determine the value of our paleo-experimental evolution system.

An example paleo-experimental evolution system

Among the various proteins so far studied by paleogeneticists, Elongation Factor-Tu (EF) is an ideal candidate for use in paleo-experimental evolution. EF is a GTP-binding protein that functions to deliver aminoacylated-tRNAs to the A-site of the ribosome and is thus an essential component of ribosome-based protein biosynthesis (Czworkowski and Moore 1996). In addition to binding all ~47 different tRNAs (at least in *E. coli*), EFs also bind to other classes of proteins such as chaperones, metabolic enzymes, structural proteins, and others (Figure 3). EFs are one of the most abundant proteins in bacteria. In addition to being a universal protein found in all known cellular life, deletion of EF is lethal (Schnell et al. 2003).

Previous studies using large protein datasets have calculated a correlation coefficient of 0.91 between environmental temperature of a host organism and the melting temperatures of a subset of a host's globular proteins (Gromiha et al. 1999). Among this subset of

proteins, EFs are known to adapt to the environmental temperature of their host organisms - EFs from thermophilic microorganisms are thermostable whereas EFs from mesophilic organisms are mesostable; supporting the notion that proteins are marginally stable (Taverna and Goldstein 2002). This suggests that a strong selective constraint shapes the thermostability profile of EF proteins.

All these properties make EF-Tu an ideal protein for a system that combines experimental evolution with synthetic biology. The combination of EF's role in cellular networks and the strong constraints acting on EF's thermostability, creates an ideal situation to knockout endogenous EF from a modern organism and replace it with an ancestral form of the protein whereby the ancestral protein shares a direct evolutionary history with the modern form of the protein. We therefore set out to generate a strain of modern bacteria (*E. coli*) in which we replaced the endogenous EF with an ancestral EF at the precise genomic location of the modern gene – thus using the modern promoter to drive expression.

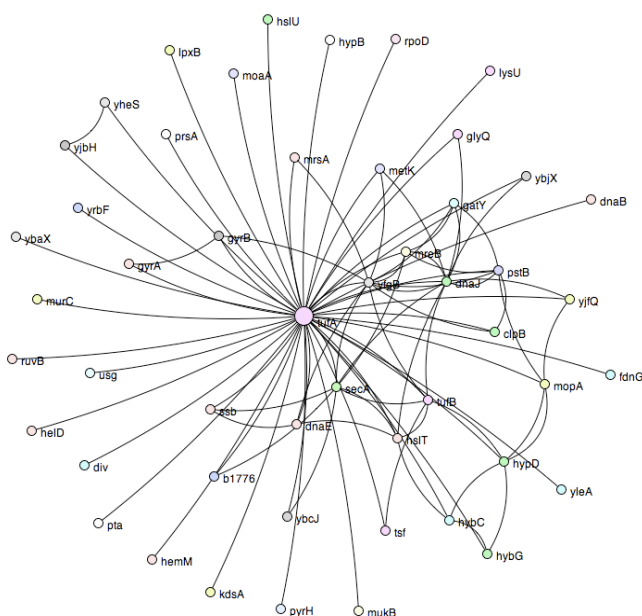


Figure 3: Bacterial EF-Tu (tufA node in center of hub) interacts with >100 cellular partners, including the ribosome, tRNAs, amino acids, GTP, EF-Ts (EF-Tu's nucleotide exchange factor) and more. This graph shows the >50 protein binding partners to EF-Tu that have been experimentally validated (binding to nucleic acids not shown). Network dataset rendered using Bacteriome.org.

To fulfill our paleo-experimental evolution objective in the laboratory, we have replaced the modern endogenous EF-Tu gene with a resurrected form of the

gene using DNA recombining technology (Datsenko and Wanner 2000). *E. coli* is unique among most bacteria in that it contains two genomic copies of EF (*tufA* and *tufB*, that differ from one another by a single amino acid). We elected to insert the ancient EF at the *tufB* genomic location since this region of the chromosome is less populated with open reading frames of other genes compared to the *tufA* location. As such, we first knocked out *tufA* and measured this effect on growth (Figure 4). As a control for comparative purposes, we also knocked out *tufB* in a separate strain to measure its effect on growth (Figure 4). Next, we precisely swapped *tufB* for an ancestral EF gene in the *tufA* knockout strain.

Our ancestral EF represents an ancestral γ -proteobacteria that is estimated to be on the order of 500 million years old (Battistuzzi et al. 2004) and has 21 out of 394 amino acids differences with *E. coli*'s *tufB*. This marks the first time an ancient gene has been genomically integrated in place of its modern counterpart within a contemporary organism. We next measured the cellular doubling time of the synthetic recombinant organism hosting the ancestral gene. Figure 4 shows that when replaced with the modern EF gene, the ancient EF gene extended the doubling time by approximately two-fold.

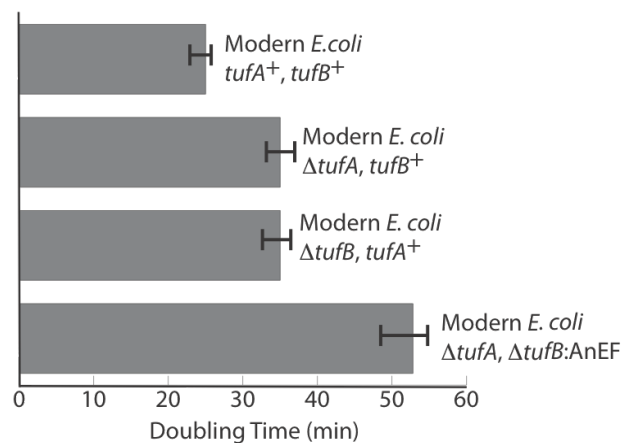


Figure 4: Precise replacement of a modern bacterial EF-Tu gene with its ~500 million year old ancestor extends the bacterial doubling time by two-fold. Two genes, *tufA* and *tufB*, (varying by just one amino acid) code for EF-Tus in *E. coli*. Precise replacement of endogenous EF-Tu requires both chromosomal *tufA* and *tufB* to be disrupted (*Schnell et al. 2003*). Deletions of *tufA* or *tufB* in the *E. coli* B strain have similar effects (~ 34 minutes) when deleted individually. The ancient EF (AnEF) has 21 (out of 392) amino acid differences with the modern EF-Tu protein. Measurements are performed in LB media at 37°C in triplicate. Modern *E. coli* B strain REL606 was obtained courtesy of R. E. Lenski (Michigan State University).

Historical contingency and the unpredictability of life

A paleo-experimental evolution system in the laboratory permits us to travel back in time to some approximation. By exploiting paleogenetics, we effectively go back in time through the history of a single component of life, capture that component, and transport it with us back to the present. In its most abstract manner, we have rewound a section of the tape of life and are giving it another opportunity to ‘evolve’ (albeit in a modern context). This approach therefore allows us to experimentally carry out Gould’s thought experiment on “replaying the tape of life” at the molecular level. We anticipate that our novel system will enable us to address long-standing questions in evolutionary and molecular biology:

- Does an organism’s history constrain its future?
- Does evolution always lead to a single and defined point or are there multiple solutions?
- How does a gene network adapt (as a whole or individual nodes)?
- Are compensatory mutations predictable?
- How do gene networks affect the evolutionary trajectory of a whole genome?
- How does selection act at the level of gene regulation vs. protein behavior?
- What is the impact of epistasis in shaping adaptive landscapes?
- Do universal biological laws govern evolution?

In addition to the points above, we anticipate that our system will enable us to address issues regarding the predictability of evolution. Along these lines, three important factors necessary to predict evolutionary outcomes are evolutionary dynamics, evolutionary rates and understanding the constraints acting on an evolving system. Changing the connectivity of a protein interaction network by swapping the network’s hub with its various evolutionary ancestors provides us with an opportunity to control some of these factors and may lead to predictability at some level. For instance, we can control the amount of stress or strain on our synthetic recombinant organism by controlling the ancestral hub we introduce into the system. Older, more ancient EFs are expected to be a greater burden when placed in a modern organism compared to an ancient EF resurrected from a node closer on a tree to the modern organism. In a system where evolutionary stressors can be controlled, how much of evolution will follow random paths? If the evolutionary trajectories are dependent on evolutionary starting points (different ancestral states), and if we can

control the factors of an evolving system; will life follow an unpredictable path?

Conclusion

In this article, we introduce and describe a novel experimental setup that we term paleo-experimental evolution. This setup weds synthetic biology with experimental evolution. The goal of this combination is to identify the historical stops along the evolutionary tracks that gave rise to modern genotypes and to explore the accessible peaks in evolutionary history, thus helping us determine the role of chance vs. necessity in evolution. Despite the unnatural properties of our laboratory system, we anticipate that our unique system will advance our ability to understand both evolutionary mechanisms and how genotype is connected to phenotype even when phenotype arises in a synthetic system.

It should be noted that our system is not limited to ancient genes. *De novo* genes can be engineered and placed in organisms as well and the evolutionary patterns that arise from their adaptation can be tested *in vivo*. Further, synthetic genes can be evolved in additional genomic backgrounds (e.g., a thermophilic and a mesophilic species) for a deeper understanding of the role that a genome’s *history* has in shaping a synthetic gene’s evolutionary trajectory when placed in a modern organism.

We anticipate that our ability to combine the two disparate fields of synthetic biology and experimental evolution will enhance our understanding of the constraints that shape biological evolution. If we are able to demonstrate that aspects of evolution are predictable regardless of whether this is due to strong selective constraints or due to historical events, this insight will be valuable in our ultimate attempts to generate artificial life and our ability to maintain (and when necessary, constrain) this life form.

Acknowledgements

This work was funded by National Aeronautics and Space Agency Astrobiology: Exobiology and Evolutionary Biology grant NNX08AO12G to E.A.G and by the NASA Astrobiology Institute through a NASA Postdoctoral Fellowship to B.K.

References

- Battistuzzi, F. U., Feijao, A. and Hedges, S. B. (2004). A genomic timescale of prokaryote evolution: insights

- into the origin of methanogenesis, phototrophy, and the colonization of land. *BMC Evol Biol* 4: 44.
- Benner, S. A., Sassi, S. O. and Gaucher, E. A. (2007). Molecular paleoscience: systems biology from the past. *Adv Enzymol Relat Areas Mol Biol* 75: 1-132, xi.
- Benner, S. A. and Sismour, A. M. (2005). Synthetic biology. *Nature Reviews Genetics* 6(7): 533-543.
- Blount, Z. D., Borland, C. Z. and Lenski, R. E. (2008). Historical contingency and the evolution of a key innovation in an experimental population of *Escherichia coli*. *Proceedings of the National Academy of Sciences of the United States of America* 105(23): 7899-7906.
- Counago, R., Chen, S. and Shamoo, Y. (2006). In vivo molecular evolution reveals biophysical origins of organismal fitness. *Molecular Cell* 22(4): 441-449.
- Czworkowski, J. and Moore, P. B. (1996). The elongation phase of protein synthesis. *Prog Nucleic Acid Res Mol Biol* 54: 293-332.
- Datsenko, K. A. and Wanner, B. L. (2000). One-step inactivation of chromosomal genes in *Escherichia coli* K-12 using PCR products. *Proceedings of the National Academy of Sciences of the United States of America* 97(12): 6640-6645.
- Dennett, D. C. (1995). Darwin's dangerous idea : evolution and the meanings of life. New York, Simon & Schuster.
- Elena, S. F. and Lenski, R. E. (2003). Evolution experiments with microorganisms: the dynamics and genetic bases of adaptation. *Nat Rev Genet* 4(6): 457-469.
- Endy, D. (2005). Foundations for engineering biology. *Nature* 438(7067): 449-453.
- Finnigan, G. C., Hanson-Smith, V., Stevens, T. H. and Thornton, J. W. (2012). Evolution of increased complexity in a molecular machine. *Nature* 481(7381): 360-364.
- Gaucher, E. A. (2007). Ancestral sequence reconstruction as a tool to understand natural history and guide synthetic biology : realizing and extending the vision of Zuckerkandl and Pauling. Oxford, Oxford University Press: 20-33.
- Gaucher, E. A., Govindarajan, S. and Ganesh, O. K. (2008). Palaeotemperature trend for Precambrian life inferred from resurrected proteins. *Nature* 451(7179): 704-707.
- Gibson, D. G., Glass, J. I., Lartigue, C., Noskov, V. N., Chuang, R. Y., Algire, M. A., Benders, G. A., Montague, M. G., Ma, L., Moodie, M. M., Merryman, C., Vashee, S., Krishnakumar, R., Assad-Garcia, N., Andrews-Pfannkoch, C., Denisova, E. A., Young, L., Qi, Z. Q., Segall-Shapiro, T. H., Calvey, C. H., Parmar, P. P., Hutchison, C. A., 3rd, Smith, H. O. and Venter, J. C. (2010). Creation of a bacterial cell controlled by a chemically synthesized genome. *Science* 329(5987): 52-56.
- Gould, S. J. (1994). The Evolution of Life On Earth. Scientific American. 271: 85-86.
- Gromiha, M. M., Oobatake, M. and Sarai, A. (1999). Important amino acid properties for enhanced thermostability from mesophilic to thermophilic proteins. *Biophysical Chemistry* 82(1): 51-67.
- Meyer, J. R., Dobias, D. T., Weitz, J. S., Barrick, J. E., Quick, R. T. and Lenski, R. E. (2012). Repeatability and contingency in the evolution of a key innovation in phage lambda. *Science* 335(6067): 428-432.
- Morris, S. C. (2003). Life's Solution: Inevitable Humans in a Lonely Universe, Cambridge University Press.
- Pauling, L. and Zuckerkandl, E. (1963). Chemical Paleogenetics Molecular Restoration Studies of Extinct Forms of Life. *Acta Chemica Scandinavica* 17: 9-&.
- Pena, M. I., Van Itallie, E., Bennett, M. R. and Shamoo, Y. (2010). Evolution of a single gene highlights the complexity underlying molecular descriptions of fitness. *Chaos* 20(2): -.
- Pennisi, E. (2011). Evolutionary biology. Evolutionary time travel. *Science* 334(6058): 893-895.
- Perez-Jimenez, R., Ingles-Prieto, A., Zhao, Z. M., Sanchez-Romero, I., Alegre-Cebollada, J., Kosuri, P., Garcia-Manyes, S., Kappock, T. J., Tanokura, M., Holmgren, A., Sanchez-Ruiz, J. M., Gaucher, E. A. and Fernandez, J. M. (2011). Single-molecule paleoenzymology probes the chemistry of resurrected enzymes. *Nature Structural & Molecular Biology* 18(5): 592-U599.
- Poelwijk, F. J., Kiviet, D. J., Weinreich, D. M. and Tans, S. J. (2007). Empirical fitness landscapes reveal accessible evolutionary paths. *Nature* 445(7126): 383-386.
- Salverda, M. L., Dellus, E., Gorter, F. A., Debets, A. J., van der Oost, J., Hoekstra, R. F., Tawfik, D. S. and de Visser, J. A. (2011). Initial mutations direct alternative pathways of protein evolution. *Plos Genetics* 7(3): e1001321.
- Schnell, R., Abdulkarim, F., Kalman, M. and Isaksson, L. A. (2003). Functional EF-Tu with large C-terminal extensions in an E-coli strain with a precise deletion of both chromosomal tuf genes. *Febs Letters* 538(1-3): 139-144.
- Stackhouse, J., Presnell, S. R., McGeehan, G. M., Nambiar, K. P. and Benner, S. A. (1990). The ribonuclease from an extinct bovid ruminant. *FEBS Lett* 262(1): 104-106.
- Taverna, D. M. and Goldstein, R. A. (2002). Why are proteins marginally stable? *Proteins-Structure Function and Genetics* 46(1): 105-109.
- Thomson, J. M., Gaucher, E. A., Burgan, M. F., De Kee, D. W., Li, T., Aris, J. P. and Benner, S. A. (2005). Resurrecting ancestral alcohol dehydrogenases from yeast. *Nat Genet* 37(6): 630-635.
- Thornton, J. W., Need, E. and Crews, D. (2003). Resurrecting the ancestral steroid receptor: ancient origin of estrogen signaling. *Science* 301(5640): 1714-1717.
- Weinreich, D. M., Delaney, N. F., DePristo, M. A. and Hartl, D. L. (2006). Darwinian evolution can follow only very

few mutational paths to fitter proteins. *Science* 312(5770): 111-114.

Wichman, H. A., Scott, L. A., Yarber, C. D. and Bull, J. J. (2000). Experimental evolution recapitulates natural evolution. *Philos Trans R Soc Lond B Biol Sci* 355(1403): 1677-1684.

Digital Evolution Exhibits Surprising Robustness to Poor Design Decisions

David M. Bryson and Charles Ofria

Department of Computer Science and Engineering
BEACON Center for the Study of Evolution in Action
Michigan State University, East Lansing, MI 48824
brysonda@egr.msu.edu

Abstract

When designing an evolving software system, a researcher must set many aspects of the representation and inevitably make arbitrary decisions. Here we explore the consequences of poor design decisions in the development of a virtual instruction set in digital evolution systems. We evaluate the introduction of three different severities of poor choices. (1) functionally neutral instructions that water down mutational options, (2) actively deleterious instructions, and (3) a lethal die instruction. We further examine the impact of a high level of neutral bloat on the short term evolutionary potential of genotypes experiencing environmental change. We observed surprising robustness to these poor design decisions across all seven environments designed to analyze a wide range challenges. Analysis of the short term evolutionary potential of genotypes from the principal line of descent of case study populations demonstrated that the negative effects of neutral bloat in a static environment are compensated by retention of evolutionary potential during environmental change.

Introduction

Since the beginning of the field, evolutionary computation has taken its inspiration from biology. Genetic algorithms (Holland, 1975), genetic programming (Koza, 1990), and evolutionary strategies (Rechenberg, 1971) all exploit the power of mutation, selection, and differential survival to generate successful solutions to complex problems. A potential drawback of these traditional evolutionary computation techniques is that all methods require the researcher to define an explicit fitness function. All of the traits desired in the solution must be explicitly accounted for within the selection regime. As the complexity of the problem increases, this requirement becomes burdensome.

To address this and other challenges, researchers are taking further inspiration from biology and leveraging natural selection as instantiated by digital evolution (McKinley et al., 2008). Self-replicating computer programs, each running on their own virtual CPU, populate these digital evolution systems. Each program can be thought of as the genome of a digital organism, and consists of a string of instructions from a pre-defined set. To produce an offspring, a digital

organism must copy its genome one line at a time, while being subject to environmental factors including other organisms and noise that causes errors (mutations) to this process. Since the digital organisms can interact and are responsible for their own replication, these systems have no explicit fitness function. The biggest power of the system is that it allows us to more easily translate concepts from natural biology. In order to direct evolution, an experimenter must craft an environment where the organisms face the same problem that the experimenter is trying to solve.

Researchers have used the Avida digital evolution system extensively to study evolutionary theory (Adami, 2006). Recent studies are pushing it into new, applied directions. For example, Knoester et al. (2007) and Beckmann et al. (2007) have explored communication and cooperation for distributed problem solving. Goldsby et al. (2007) investigate digital evolution as a tool for evolving software models for dynamic systems. Grabowski et al. (2008, 2010, 2011) study the evolution of movement and decision making. Many of these new experimental directions require changes to the virtual hardware and instruction set to support interaction with the environment and enhance the success of evolved solutions. The design of the instruction set architecture within an evolvable system can play an important role in the robustness and adaptability of evolved solutions (Ofria et al., 2002).

Changes to the instruction set may have a profound effect on the evolutionary potential of the system with respect to the environment. It is difficult, if not impossible, to assess the impact of instruction set changes a priori. A seemingly beneficial change may in fact have unintended negative interactions with other aspects of the system. Here we have investigated three types of poor instruction set design decisions, functionally neutral instructions that bloat the instruction set, actively deleterious instructions that poison the organism, and a lethal instruction. We evaluate the evolutionary potential of each instruction set given a fixed amount of evolutionary time. In order to test the broad effect of each modification, we crafted seven computational environments representing a wide range of desired capabilities. We evalu-

ate the final results of experiments performed in each environment with each instruction set modification.

Given a fixed environment, a particular instruction set may show greater evolutionary potential in comparison to another. However, it is possible that aspects of an instruction set may demonstrate better adaptability to changing circumstances. As evolution progresses, organism genomes lock in features and genetic organization that are beneficial. The structure of these genomes impact their potential when the environment changes. We investigate this by evaluating short term evolutionary potential of genomes following environmental change. We examine how this potential changes relative to the progress of evolution in the origin environment.

Methods

We performed all experiments using Avida version 2.12.3¹. We tested each instruction set architecture with 200 replicate populations in each of seven computational environments. We evolved the populations on a structured 60 x 60 toroidal grid for 100,000 updates². Organisms were subject to a mutation rate of 2.5×10^{-3} per site in the genome, along with a 0.5×10^{-3} probability each for a single instruction insertion or deletion per site in the genome. Given that the ancestral organism had a length 100 genome, its mutation load was an average of 0.35 mutations per offspring, though size changes would change this load over time. All mutations, insertions, and deletions occurred upon division of the offspring. We seeded each population with a full complement of 3600 organisms with an ancestral genotype capable only of self-replication.

Instruction Sets

The HEADS architecture in Avida is the default virtual CPU configuration. The virtual hardware that implements this instruction set contains 26 commands designed to operate on a genomic program. It has three 32-bit registers, two stacks, four heads that point to positions in the genome, and input and output buffers. Among the 26 instructions in the set are three no-operation instructions, which can serve to modify the default behavior of other instructions, five flow-control instructions, three conditional instructions, seven arithmetic and logic instructions, five data movement instructions, and three instructions for self-replication.

The BLOAT instruction sets test what happens if you add too many useless, although not directly disruptive, instructions to the instruction set. They extend the HEADS architecture with the addition of one or more copies of the no-operation instruction, `nop-X`, which is functionally neu-

tral, in that it does not alter the state of the virtual CPU. Additionally, unlike the three default no-operation instructions, it does not alter the behavior of other instructions. We tested four BLOAT instruction sets, varying the mutational frequency of the `nop-X` instruction. BLOAT-1 adds `nop-X` with a frequency of 1, yielding an effective mutational frequency of 0.037 for each instruction. BLOAT-3, BLOAT-10, and BLOAT-30 each increase the frequency of `nop-X` to 3, 10, and 30, respectively. In BLOAT-30, the effective mutational frequency of the `nop-X` instruction was 0.536, with 0.018 for each of the remaining 26 standard instructions.

With the POISON instruction sets we are testing what happens when we make a poor decision by adding an instruction that can actually disrupt the functionality of the organisms upon execution. These instruction sets extend the HEADS architecture with the addition of a `poison` instruction that, when executed, reduces the metabolic rate of the organism by a configurable severity. Reduced metabolic rate translates to fewer relative CPU cycles, and therefore diminished competitive ability. We tested three poison severities, 0.003, 0.01, 0.03, which reduce metabolic rate by 0.3%, 1%, and 3% each time the organism executes the instruction. We hypothesized that lower penalties might be more detrimental to long term evolutionary, because they may slip in and accumulate over time.

Lastly, the DIE instruction set sought to determine what happens when we make a catastrophic error in including an instruction in the set. This instruction set adds a single `die` instruction to the HEADS architecture. The presence of a `die` instruction in a genome is not itself lethal. If the organism executes the instruction during its lifetime, however, the organism will be immediately removed from the population.

Environment	Rewarded Functions
Logic-9	Nine 1- and 2-input logic operations.
Logic-77	Seventy-seven 1-, 2-, and 3-input logic operations.
Match-12	Generate up to 12 specific numbers.
Fibonacci-32	Output up to 32 numbers of the Fibonacci sequence, in order.
Sort-10	Input 10 random numbers and output in correctly sorted order.
Limited-9	Logic-9 environment with a limited resource associated with each task.
Navigation	Successfully traverse a labeled pathway.

Table 1: The seven environments used to test instruction set modifications.

¹Avida 2.12.3 source code is available for download, without cost, from <http://avida.devosoft.org/>. Specific instruction set configurations used are available upon request.

²An update is the natural unit of time in Avida, equal to an average of 30 instructions executed per living organism.

Environments

Avida supports a wide range of computational environments. We used seven distinct environments (Table 1), each of which focuses on a different aspect of the virtual architecture and presents unique evolutionary pressures. Activities (or tasks) whose performance provide a metabolic reward define the environment. These rewards increase the computation speed of the digital organism's virtual CPU, making it possible to obtain a competitive advantage relative to other organisms in the population.

The *Logic-9* environment consists of metabolic rewards for all possible 1- and 2-input binary logic operations; there are 9 unique operations after removing symmetries and excluding trivial operations. The environment rewards the performance of these tasks multiplicatively, thus virtual CPU speed will increase exponentially as the organism performs additional tasks. There are five reward levels associated with groups of logic operations, ranked by difficulty. The easiest group (NOT and NAND) will double computational speed, while the highest level (EQU) increases execution speed by thirty-two times. The environment rewards each task only once during an organisms' lifetime. This environmental setup is the default for Avida and many previous experiments have used it. (Lenski et al., 1999, 2003; Misevic et al., 2006)

The *Logic-77* environment increases the size and complexity of the *Logic-9* environment by adding a reward for all unique 3-input binary logic operations. In contrast to the *Logic-9* environment, all operations provide an equal benefit, doubling the execution speed of the virtual CPU for the first time the organism performs computation. Yedid et al. (2009) used this environment.

The *Match-12* environment tests organisms' ability to build arbitrary numbers, a task that we have previously observed to constitute an obstacle to evolution (unpublished data). The environment grants rewards in an additive manner for outputting each of twelve possible numbers, unrelated to the random inputs. We selected numbers spaced approximately exponentially throughout the 32-bit number space, but the numbers contain no explicit patterns to them. The environment rewards the output of each number only once during an organism's lifetime. Output evaluation allows near matches, but the reward decays via a half-life function based upon the number of bits that are incorrect with a minimum threshold of 22 bits correct to prevent most numbers from triggering many 'lucky guesses'.

The *Fibonacci-32* environment rewards organisms multiplicatively for each number in the Fibonacci sequence until the 32nd iteration of the sequence. After this target, the environment penalizes the organism at half this rate for additional numbers output, whereby outputting 64 additional numbers will effectively negate all benefit of the first 32. The purpose of this setup is to examine the capability of an instruction set to support finite recursion and conditional

looping.

The *Sort-10* environment supplies a list of 10 random inputs, and rewards organisms for outputting those values in descending order. Similar to the *Match* environment, the reward value decays via a half-life function for each incorrectly sorted value, based on the number of moves required to shift it to the correct order. Given the limited number of available registers, this task requires the use of the stacks and relatively complex flow control.

The *Limited-9* environment, based on the *Logic-9* environment, offers metabolic rewards for all possible 1- and 2-input binary logic operations. However, unlike the *Logic-9* environment, the *Limited-9* environment associates a separate, consumable resource with each task, the amount of which determines the exact reward value. Each resource flows into the environment at a rate of 100 units per update, and out at 1% of the remaining concentration. If no organisms are using the resource it will level out to 10,000 units. This environment was first used in Cooper and Ofria (2003).

The *Navigation* environment rewards organisms for successfully navigating a circuitous path marked by cues ("sign posts") including "turn left", "turn right", and "repeat last turn", as described in Grabowski et al. (2010) This task requires the use of basic memory, looping, and decision making. Additionally, the environment tests robustness of instruction set architectures to the addition of several experiment-specific instructions, in this instance for sensing and moving in the virtual maze.

Short Term Evolutionary Potential (STEP) Sampling

Short-term evolutionary potential (STEP) sampling explores the mid-range fitness landscape of a reference genotype by evolving repeated short runs from the same starting point and analyzing aggregate statistics of the outcome of each. This procedure involves injecting the reference genotype as a single organism in an otherwise empty experimental world configured similarly to the settings used in the experiment that was the source of the genotype. We then allow the world to evolve for a short period, 10,000 updates (approximately 1,000 generations) for the work presented here, after which we collect metrics of interest, such as phenotype and fitness. We repeat this procedure with the same reference genotype multiple times for statistical assessment of the genotype's evolutionary potential.

General Performance Evaluation

We have focused on two measures of evolved populations to evaluate the general performance of each instruction set architecture: mean fitness and task success. Both measure ability of the evolved organisms to perform tasks within the environment.

Mean fitness averages the fitness values of each living organism in the population at the moment the experiment fin-

	Logic-9	Logic-77	Match-12	Fibonacci-32	Sort-10	Limited-9	Navigation
<i>Heads</i>	19.71 (19.33, 19.81)	15.22 (12.81, 16.34)	0.198 (0.159, 0.243)	3.645 (3.299, 4.082)	-0.482 (-0.598, -0.324)	4.220 (4.095, 4.294)	1.447 (1.103, 3.197)
<i>Bloat-1</i>	19.33 (18.12, 19.72)	14.42 (12.36, 15.89)	0.211 (0.180, 0.242)	3.241 (3.010, 3.815)	-0.498 (-0.608, -0.372)	4.301 (4.141, 4.418)	1.123 (1.051, 2.453)
<i>Bloat-3</i>	19.67 (18.15, 19.75)	12.31 (10.71, 14.37)	0.176 (0.142, 0.207)	3.400 (3.206, 3.762)	-0.538 (-0.611, -0.441)	4.247 (4.138, 4.375)	1.123 (1.031, 3.202)
<i>Bloat-10</i>	14.80 (14.73, 17.32)	11.77 (10.66, 14.32)	0.106 (0.082, 0.127)	2.621 (2.540, 3.116)	-0.696 (-0.717, -0.675)	4.526 (4.312, 4.779)	1.068 (1.022, 2.640)
<i>Bloat-30</i>	14.38 (13.70, 14.61)	7.74 (7.69, 8.67)	-0.053 (-0.170, 0.083)	1.768 (1.722, 1.802)	-0.770 (-0.785, -0.741)	4.480 (4.317, 4.684)	2.872 (1.313, 3.165)
<i>Poison-0.003</i>	19.57 (18.72, 19.73)	14.11 (12.19, 15.96)	0.206 (0.177, 0.252)	4.240 (3.496, 4.623)	-0.342 (-0.483, -0.225)	4.152 (4.071, 4.226)	1.157 (1.056, 1.635)
<i>Poison-0.01</i>	19.41 (18.20, 19.76)	12.63 (11.64, 14.76)	0.159 (0.137, 0.214)	3.309 (3.181, 3.825)	-0.591 (-0.664, -0.442)	4.300 (4.127, 4.492)	1.342 (1.052, 3.386)
<i>Poison-0.03</i>	18.66 (17.79, 19.62)	12.19 (11.42, 14.40)	0.157 (0.130, 0.192)	3.417 (3.219, 3.883)	-0.464 (-0.616, -0.313)	4.290 (4.207, 4.538)	1.356 (1.068, 3.275)
<i>Die</i>	19.60 (17.84, 19.78)	14.28 (11.93, 16.32)	0.181 (0.156, 0.235)	3.476 (3.211, 3.909)	-0.581 (-0.634, -0.438)	4.422 (4.288, 4.563)	1.324 (1.060, 3.150)

Table 2: Fitness results for all 8 test instruction sets and the HEADS control architecture. Each entry shows the median \log_2 population mean fitness in the respective environment, with 95% confidence intervals in parentheses. Bold entries indicate significant ($p < 0.05$) deviations after sequential Bonferroni correction.

	Logic-9	Logic-77	Match-12	Fibonacci-32	Sort-10	Limited-9	Navigation
<i>Heads</i>	0.842 (0.835, 0.847)	0.207 (0.179, 0.227)	0.145 (0.144, 0.146)	0.205 (0.177, 0.237)	1.42×10^{-4} (1.14, 1.71)	0.906 (0.896, 0.912)	4.35×10^{-3} (3.98, 7.54)
<i>Bloat-1</i>	0.835 (0.753, 0.843)	0.198 (0.173, 0.218)	0.146 (0.145, 0.147)	0.177 (0.174, 0.207)	1.46×10^{-4} (1.10, 1.67)	0.906 (0.896, 0.911)	4.05×10^{-3} (3.96, 5.84)
<i>Bloat-3</i>	0.839 (0.825, 0.846)	0.171 (0.150, 0.197)	0.146 (0.145, 0.147)	0.203 (0.176, 0.236)	1.37×10^{-4} (1.14, 1.54)	0.899 (0.829, 0.911)	3.99×10^{-3} (3.97, 7.20)
<i>Bloat-10</i>	0.747 (0.744, 0.750)	0.166 (0.150, 0.203)	0.146 (0.144, 0.146)	0.174 (0.149, 0.178)	1.01×10^{-4} (0.99, 1.04)	0.832 (0.823, 0.894)	4.00×10^{-3} (3.97, 6.86)
<i>Bloat-30</i>	0.736 (0.648, 0.744)	0.114 (0.113, 0.126)	0.146 (0.125, 0.147)	0.120 (0.119, 0.120)	9.7×10^{-5} (9.6, 9.9)	0.777 (0.732, 0.808)	7.57×10^{-3} (4.08, 7.82)
<i>Poison-0.003</i>	0.841 (0.828, 0.846)	0.194 (0.172, 0.217)	0.147 (0.146, 0.148)	0.239 (0.205, 0.285)	1.67×10^{-4} (1.45, 1.97)	0.910 (0.903, 0.915)	3.99×10^{-3} (3.97, 4.79)
<i>Poison-0.01</i>	0.839 (0.821, 0.845)	0.174 (0.162, 0.204)	0.146 (0.145, 0.146)	0.179 (0.176, 0.208)	1.22×10^{-4} (1.07, 1.54)	0.898 (0.844, 0.909)	4.33×10^{-3} (3.97, 7.77)
<i>Poison-0.03</i>	0.838 (0.773, 0.844)	0.170 (0.159, 0.202)	0.146 (0.145, 0.147)	0.202 (0.177, 0.237)	1.52×10^{-4} (1.11, 1.82)	0.911 (0.897, 0.916)	4.33×10^{-3} (3.97, 7.97)
<i>Die</i>	0.827 (0.754, 0.841)	0.198 (0.163, 0.224)	0.146 (0.145, 0.147)	0.195 (0.176, 0.210)	1.27×10^{-4} (1.04, 1.52)	0.906 (0.840, 0.913)	4.29×10^{-3} (3.97, 6.58)

Table 3: Task success results for all 8 test instruction sets and the HEADS control architecture. Each entry shows the median normalized task success in the respective environment, with 95% confidence intervals in parentheses. Bold entries indicate significant ($p < 0.05$) deviations after sequential Bonferroni correction.

ished. It takes into account both the computational capability of the organism and the efficiency of self-replication, also called the "gestation time". We examined the distributions of these fitness values for all instruction set variants in each environment. For each modified instruction set, we compared the 200 population fitness values with those of the control (HEADS) instruction set architecture using a Wilcoxon rank-sum test. We determined significance using $\alpha = 0.05$ with sequential Bonferroni correction. Confidence intervals, as shown in tables below, represent 2.5% and 97.5% quantiles that we generated using non-parametric bootstrap with 10,000 iterations.

Task success is a direct examination of the computational capabilities of the organisms within the final population, for the specific environment of the experiment. We measure the task success of a population as the sum of the qualities by which the average organism performs each task. To calculate a task success t_p of population p , we determine each organism's quality at each task and then sum over these values, finally dividing by the total number of organisms in the population. More formally,

$$t_p = \sum_{i=1}^{N_p} \sum_{j=1}^T \frac{q_{i,j}}{N_p} \quad (1)$$

where N_p is the number of organisms in population p , T is the number of tasks in the environment, and $q_{i,j}$ is the quality q at which organism i is performing task j . Task quality (q) is a value between 0 and 1, where 1 means the organism has found a perfect solution for a task. Environments that support near-matches use task quality to adjust the metabolic reward accordingly. The maximum task success for a given environment is equal to the total number of tasks rewarded in that environment; for example the maximum task success of the Logic-9 environment is 9. Normalized task success, as presented in the following results, divides the observed task success by the maximum in each environment. Similar to population mean fitness, we compared the distribution of task success of each instruction set to the control architecture using a Wilcoxon rank-sum test, sequential Bonferroni correction, and non-parametric bootstrap confidence intervals.

Upon analysis, all three POISON instruction sets and the DIE instruction set demonstrated no significant variation in either population mean fitness (Table 2) or task success (Table 3) across all seven environments. Indeed, the distributions of observed results were largely similar, regardless of the severity of the penalty associated with a given instruction. Likewise, the BLOAT-1 and BLOAT-3 instructions sets showed comparable performance to the HEADS control. This would indicate that a single poor choice of an instruction, no matter how bad, is not likely to significantly limit evolutionary outcomes.

The BLOAT-10 and BLOAT-30 instruction sets, represent-

ing high levels of instruction mutational dilution, demonstrate some negative effects in final population performance. In the Logic-9 environment, both BLOAT-10 and BLOAT-30 showed significantly decreased fitness and task success. Runs with these instruction sets evolved one fewer task on average in the Logic-9 environment when compared to the control. The Logic-77 environment similarly demonstrated decreased fitness and task success, but unlike in the Logic-9 environment, the BLOAT-30 instruction set was notably worse than BLOAT-10. This pattern of declining performance was also observed in the Fibonacci-32 environment, with task success indicating that the populations are outputting fewer one to three fewer numbers in the sequence as instruction set dilution increases.

The Limited-9 environment demonstrated a split between fitness and task success results for BLOAT-10 and BLOAT-30. The fitness results for both instruction sets were greater than the HEADS control, though not significantly after Bonferroni correction. The BLOAT-10 instruction set demonstrated task success that was somewhat, though not significantly, reduced compared to the control. Task success with the BLOAT-30 instruction set, however, was significantly reduced, with populations typically evolving one fewer task, as compared to the HEADS instruction set.

In the Navigation environment, the BLOAT-10 instruction set demonstrated comparable performance to the control for both fitness and task success. The BLOAT-30 instruction, on the other hand, showed significantly improved median fitness. Task success was also notably increased, nearly double all other instructions sets, though not statistically significant from the control after Bonferroni correction. Despite the increase, the populations are still quite far from exploiting the opportunities in this environment, taking advantage of less than 1% of the potential resources.

The Match-12 environment showed no variation in task success with any of the tested instruction sets. The BLOAT-10 and BLOAT-30 instruction sets both demonstrated significantly lower fitness, with BLOAT-30 the most severely depressed. Given the lack of variation in task success, these fitness result likely reflect the impact of neutral instruction set bloat on the evolution of replication efficiency in these digital organisms.

Lastly, the Sort-10 environment was significantly reduced in both fitness and task success under both BLOAT-10 and BLOAT-30. The differences observed, however, were relatively insubstantial. None of the tested instruction sets, including the HEADS control, were able to take advantage of the opportunities in the Sort-10 environment; all sets demonstrated $\ll 1\%$ of the potential task success. The current limitations of the virtual CPU appear to make this task incredibly difficult to evolve.

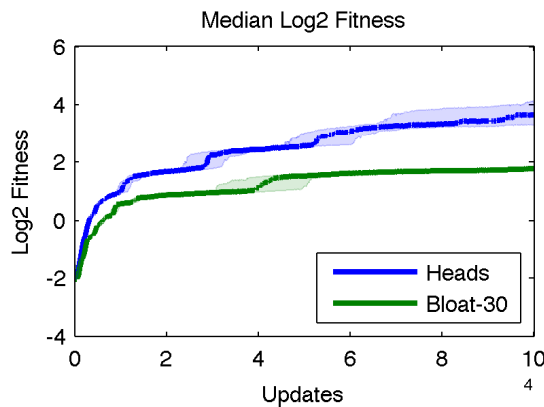


Figure 1: Median \log_2 fitness trajectory of the HEADS (blue line) and BLOAT-30 (green line) architectures. Lines calculated from all 200 replicates of each instruction set in the Fibonacci-32 environment. Shaded regions show 95% bootstrap confidence intervals, 10,000 iterations.

Impact on Evolutionary Potential

The Bloat instruction sets, especially BLOAT-10 and BLOAT-30, showed reduced performance when examining a fixed end point, as shown above. The fitness trajectories of these instruction sets demonstrated a corresponding drag on evolution throughout the entire history of the runs, relative to the HEADS control instruction set (see Figure 1 for an example). Despite this apparent drag, fitness was still rising at the end of the experiment, albeit more slowly.

The neutral bloat represented by the Bloat instruction sets, although detrimental to the rate of evolution, may have a beneficial effect on the genetic architecture of the evolved genomes. The nop instructions will tend to decouple strings of other instructions, such that genetic functions must be more loosely coupled. This property may afford greater evolutionary potential when the genomes experience environmental change. We have tested this hypothesis by performing STEP sampling of genotypes in a new environment, never before encountered in the history of the genotype.

We extracted the principal line of descent, the complete lineage of ancestral genotypes that gave rise to the final, numerically dominant genotype, from 12 selected runs utilizing the HEADS and BLOAT-30 instruction sets that we initially evolved in the Logic-9 and Fibonacci-32 environments. These two environments both demonstrated variation in performance between the HEADS and BLOAT-30 instruction sets, and present computationally unique challenges to the organisms (logic computation in Logic-9 and loop coordination in Fibonacci-32). The genotypes along each of the lines of descent were STEP sampled, still using their native instruction set, but in the opposite environment. For example, we placed genotypes evolved in the Logic-9 environ-

ment into the Fibonacci-32 environment and evolved them for 10,000 updates. We sampled each genotype ten times and examined the fitness and task success of the resulting populations.

The STEP sampling results of the HEADS control instruction set show that the short term evolutionary potential of the genotypes, declined in the Logic-9 environment as evolution progressed in the Fibonacci-32 environment (see Figure 2). All sampled lines of descent with the HEADS architecture demonstrate similar patterns of evolutionary potential in the Fibonacci-32 to Logic-9 shift. The BLOAT-30 instruction set runs, on the other hand, show a relatively flat trend of evolutionary potential. Additionally, the BLOAT-30 instruction set demonstrated an increased number of high potential outliers throughout all of the sampled lines of descent originally evolved in the Fibonacci-32 environment (see Figure 4).

Both the HEADS control and the BLOAT-30 demonstrated a consistent pattern of gradual decline in short term evolutionary potential when sampling genotypes originally evolved in the Logic-9 source environment within the Fibonacci-32 sample environment (see Figure 3). Similar to the Fibonacci-32 to Logic-9 environment transition, the BLOAT-30 instruction set showed notably more outlier samples of high potential. However, the overall spread and trend of samples of the BLOAT-30 genotypes were comparable to the HEADS instruction set.

In order to assess the generality of the observed patterns, we STEP sampled the final dominant genotype from all 200 runs of each of the original HEADS and BLOAT-30 experiments from the Logic-9 and Fibonacci-32 environments in the appropriate alternate environment. As observed in the line of descent sampling, the BLOAT-30 instruction set genotypes from the Fibonacci-32 environment demonstrated significantly greater potential ($p < 0.017$; Wilcoxon rank-sum test) when sampled in the Logic-9 environment (median \log_2 fitness 10.40) in comparison to genotypes evolved with the HEADS instruction set (median \log_2 fitness 9.653). The transition from the Logic-9 environment to the Fibonacci-32 environment showed the opposite results, with the HEADS instruction set resulting in significantly greater ($p < 0.026$) evolutionary potential (median \log_2 fitness 1.836; Wilcoxon rank-sum test) in comparison to the BLOAT-30 instruction set (median \log_2 fitness 1.768).

Discussion

In examination of general performance, evolution demonstrated surprising robustness to increasingly poor design decisions. The addition of individual instructions that were incredibly deleterious or lethal made no significant difference in the evolutionary potential of the system across a wide range of static test environments. Similarly, low levels of neutral instruction set bloat contributed negligibly to the observed performance. These results indicate that dig-

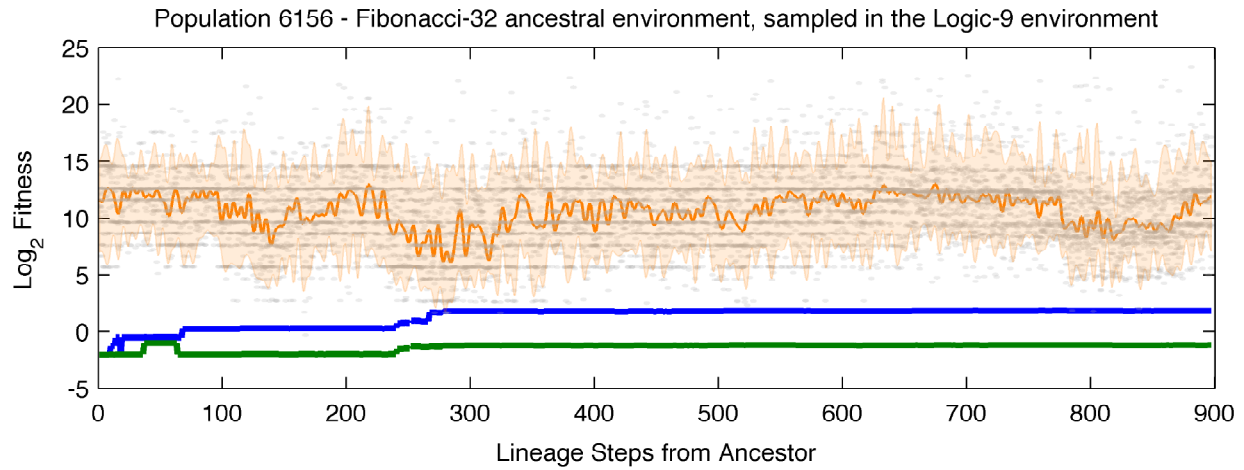


Figure 2: STEP sampling results of population 6156 originally evolved in the Fibonacci-32 environment with the HEADS instruction set. Blue line: fitness of the reference genotype in the Fibonacci-32 environment. Green line: initial fitness of the reference genotype in the Logic-9 environment. Orange line: median STEP results, smoothed using a FFT, shown with 95% quantiles. Gray circles: individual STEP results. The first point (step 0) shows the performance of the ancestral genotype.

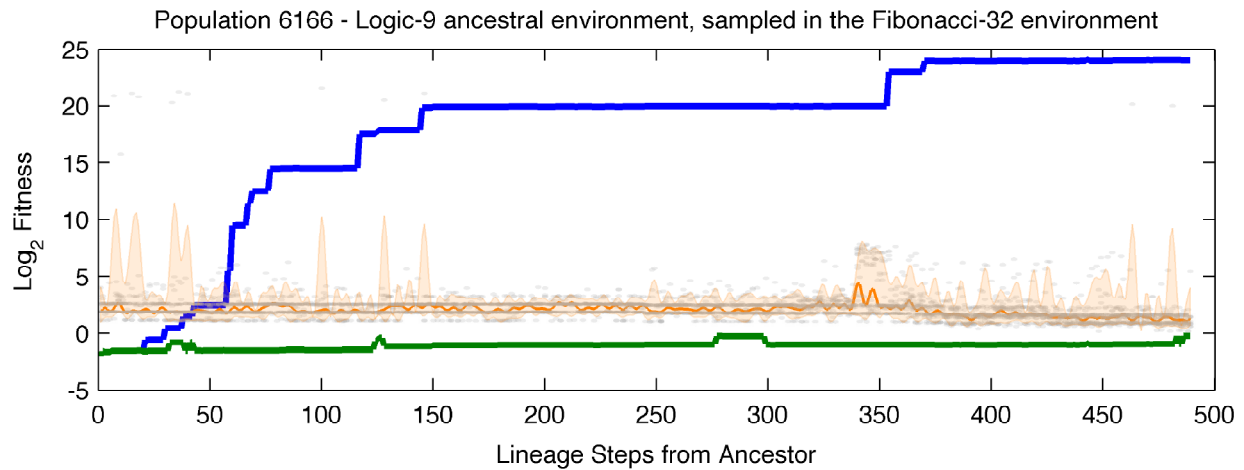


Figure 3: STEP sampling results of population 6166 originally evolved in the Logic-9 environment with the HEADS instruction set. Blue line: fitness in the Logic-9 environment. Green line: initial fitness in the Fibonacci-32 environment. Orange line: median STEP results (smoothed) with 95% quantiles. Gray circles: individual STEP results.

ital evolution can reasonably overcome individual or small sets of detrimental design decisions, regardless of the severity of the error. Populations exhibit substantial declines only if many poor decisions compound on one another.

High levels of instruction set bloat, diluting the frequency of functional mutations, resulted in an overall significant drag on evolution. Despite this dilution decreasing the rate of evolution, populations were still gaining fitness and task success, indicating that evolution could potentially overcome the detrimental effects of such poor designs given ad-

ditional time. Although the BLOAT-30 performed poorly in the initial experiments, STEP sampling showed that, under certain circumstances, the increased proportion of neutral mutations associated with instruction set bloat can actually improve evolutionary potential when changing the environment. The genetic architecture of the genotypes from the Fibonacci-32 environment with the Bloat-30 instruction set, broken up by neutral instructions, showed to be more adaptable to the logic flow necessary for success in the Logic-9 environment. Conversely, the genotypes evolved with the

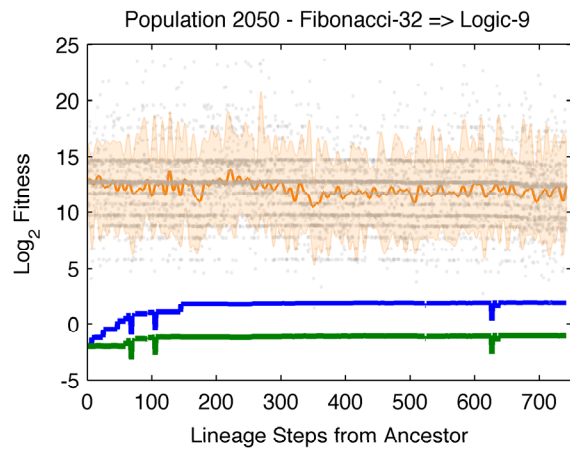


Figure 4: Fitness STEP sampling results of population 6166 originally evolved in the Logic-9 environment with the BLOAT-30 instruction set. Blue line: fitness in the Logic-9 environment. Green line: initial fitness in the Fibonacci-32 environment. Orange line: median STEP results (smoothed) with 95% quantiles. Gray circles: individual STEP results.

Bloat-30 instruction set in the Logic-9 environment performed worse in the Fibonacci-32 environment, indicating that the looping structures necessary in the Fibonacci-32 environment likely benefit from more closely connected instructions.

Acknowledgements

The authors wish to thank Heather Goldsby for helpful discussion and Richard Lenski for guidance in developing STEP sampling. This material is based in part upon work supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454 and Grant CCF-0643952. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Adami, C. (2006). Digital genetics: unravelling the genetic basis of evolution. *Nature Reviews Genetics*, 7(2):109–118.

Beckmann, B., McKinley, P. K., Knoester, D. B., and Ofria, C. (2007). Evolution of Cooperative Information Gathering in Self-Replicating Digital Organisms. In *Proceedings of the First IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, Boston, Massachusetts.

Cooper, T. F. and Ofria, C. (2003). Evolution of stable ecosystems in populations of digital organisms. In Standish, R. K., Bedau, M. A., and Abbass, H. A., editors, *Artificial Life VIII: Proceedings of the Eighth International Conference on Artificial life*, pages 227–232, Cambridge, MA. International Society of Artificial Life, MIT Press.

Goldsby, H. J., Knoester, D. B., Cheng, B. H. C., McKinley, P. K., and Ofria, C. A. (2007). Digitally Evolving Models for Dynamically Adaptive Systems. In *Proceedings of the ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, Minneapolis, Minnesota.

Grabowski, L. M., Bryson, D. M., Dyer, F. C., Pennock, R. T., and Ofria, C. (2011). Clever Creatures: Case Studies of Evolved Digital Organisms. In *Advances in Artificial Life, ECAL 2011: Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems*, pages 276–283. MIT Press.

Grabowski, L. M., Bryson, D. M., Pennock, R. T., Dyer, F., and Ofria, C. (2010). Early evolution of memory usage in digital organism. In *Proceedings of the 12th International Conference on the Synthesis and Simulation of Living Systems*, pages 224–231.

Grabowski, L. M., Elsberry, W. R., Ofria, C. A., and Pennock, R. T. (2008). On the Evolution of Motility and Intelligent Tactic Response. In *GECCO '08: Proceedings of the 2008 conference on Genetic and evolutionary computation*.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI.

Knoester, D. B., McKinley, P. K., Beckmann, B., and Ofria, C. A. (2007). Directed Evolution of Communication and Cooperation in Digital Organisms. In *Proceedings of the 9th European Conference on Artificial Life*, Lisbon, Portugal. Springer.

Koza, J. R. (1990). Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems. Technical Report STAN-CS-90-1314.

Lenski, R., Ofria, C., Pennock, R. T., and Adami, C. (2003). The Evolutionary Origin of Complex Features. *Nature*, 423:139–144.

Lenski, R. E., Ofria, C., Collier, T. C., and Adami, C. (1999). Genome complexity, robustness and genetic interactions in digital organisms. *Nature*, 400(6745):661–664.

McKinley, P. K., Cheng, B. H. C., Ofria, C., Knoester, D., Beckmann, B., and Goldsby, H. (2008). Harnessing Digital Evolution. *IEEE Computer*, 41(1).

Misevic, D., Ofria, C., and Lenski, R. E. (2006). Sexual reproduction shapes the genetic architecture of digital organisms. *Proceedings of the Royal Society of London: Biological Sciences*, 273:457–464.

Ofria, C., Adami, C., and Collier, T. (2002). Design of Evolvable Computer Languages. *IEEE Transactions on Evolutionary Computation*, 6(4):420–424.

Rechenberg, I. (1971). *Evolutionstrategie: Optimierung technischer Systeme und Prinzipien der biologischen Evolution*. PhD thesis, Berlin Technical University.

Yedid, G., Ofria, C. A., and Lenski, R. E. (2009). Selective Press Extinctions, but Not Random Pulse Extinctions, Cause Delayed Ecological Recovery in Communities of Digital Organisms. *The American Naturalist*, 173(4):E139–E154.

The role of deleterious mutations in the adaptation to a novel environment

Arthur W. Covert III^{1,2*}, Jared Carlson-Stevermer^{1,2}, Dakota Z. Derrberry^{1,2}, and Claus O. Wilke^{1,2}

¹ Section of Integrative Biology, Center for Computational Biology and Bioinformatics, and Institute for Cellular and Molecular Biology, The University of Texas at Austin.

² BEACON, Center for the Study of Evolution in Action,

*Author to whom correspondence should be sent

Art.Covert@austin.utexas.edu

Abstract

Organisms adapt by accumulating beneficial mutations. Yet sometimes these beneficial mutations are not directly accessible, and organisms may have to cross a fitness valley before further adaptation is possible. A few recent works have shown that crossing of fitness valleys, as evidenced by fixation of deleterious mutations, may be surprisingly common in adaptation, and may be an important contributor to long-term fitness increase. Here we ask how important crossing of fitness valleys is for organisms that have reached a local fitness peak in one environment and are then placed into a new environment. We compare two treatments of evolving digital organisms, one in which organisms are exposed to deleterious mutations and thus can freely explore fitness valleys, and one in which they are prevented from experiencing deleterious (but not lethal) mutations and thus cannot. We find that organisms that are exposed to deleterious mutations always do at least as well as organisms that are not. Whether organisms exposed to deleterious mutations do better depends on the relative similarity and complexity of the old and new environment. We conclude that crossing of fitness valleys is important for successful adaptation to certain types of novel environments.

Introduction

Conventional wisdom holds that beneficial mutations are good and deleterious mutations are bad. Yet in a finite population, deleterious mutations can contribute to the long-term evolutionary success by allowing the population to traverse a fitness valley leading to a higher fitness peak (Weissman et al. 2010, van Nimwegen and Crutchfield 2000). Therefore, deleterious mutations cannot be unconditionally bad. There must be well-defined scenarios under which a hypothetical population experiencing no deleterious mutations would fare worse than a population that does experience them.

Indeed, several recent works have highlighted that crossing fitness valleys may be important for long-term evolutionary success. Simulations of RNA folding have found transitions between fitness peaks in the form of “fitness reversal” of deleterious mutations (Cowperthwaite et al 2007). Experiments with self-replicating computer programs have shown that such fitness reversals may actually open up new areas of the fitness landscape, areas which were inaccessible except via a deleterious mutation (Covert 2010, Lenski et al

2003). Experiments in *Saccharomyces cerevisiae* have uncovered at least one instance of a fitness reversal in an organic system (Kvitek and Sherlock 2011). This finding suggests that the fitness landscape of yeast is rugged, and that it requires at least the occasional valley crossing via a fitness reversal.

However, some authors have conjectured that a change to a novel environment can eliminate the need for crossing fitness valleys (Whitlock 1997, Whitlock et al 1995). A sufficiently large environmental change may turn a high-fitness peak into low-lying region in the fitness landscape, from which there are many new ways to climb up. If an environmental change creates a large number of new adaptive opportunities, a population should be able to adapt without having to cross any fitness valleys. Fixation of beneficial mutations alone should drive the population towards new fitness peaks. On the other hand, if the environmental change results in a rugged fitness landscape, then high-fitness regions may still only be accessible by traversal of fitness valleys, and deleterious mutations may be required for successful adaptation.

Here, we test the importance of crossing fitness valleys in the adaptation of digital organism. Digital organisms are self-replicating computer programs that evolve to perform various logical functions (corresponding to phenotypic traits). We can manipulate environmental complexity by changing how many and which logical functions are rewarded (Lenski et al 1999). We also can monitor all mutations as they appear in the population, and prevent mutations with certain characteristics (such as deleterious mutations) from ever entering the population. This setup allows us to directly compare the evolution of populations experiencing and not experiencing deleterious mutations (Covert 2010). We find that deleterious mutations are most important for long-term evolutionary success if the new environment rewards a small number of new traits that are complex and difficult to evolve. By contrast, if the new environment rewards either a large number of new traits or a small number of new traits that are less complex, deleterious mutations provide less benefit.

Methods

Experimental system. We used the digital life system Avida, version 2.12.2, for all experiments (Ofria and Wilke 2004). In Avida, digital organisms evolve and adapt to perform various one and two input logical functions (Table 1). Populations are

normally seeded with organisms that can do nothing but self-replicate. The code that makes up a digital organism is composed of simple CPU instructions, which are colloquially referred to as the genome. Mutation acts on these genomes by changing one instruction to a new randomly chosen instruction. Over time, these organisms evolve to do logical tasks that reward them with more CPU cycles they can use to execute their genomes more rapidly. Thus, they metabolize inputs from the environment to perform logical functions that give them additional energy to self-replicate more quickly.

All experiments were done with a mutation rate of 25% on divide and all populations were allowed to grow to a maximum size of 10,000 organisms.

Adaptations in one- and two-trait environments. Seed organisms were generated by adapting populations to one logic function, but not others. We called these initial replicates “priming populations” and began them with 50 replicates of populations with a standard seed organism that could do nothing but self-replicate. The replicate populations were first evolved for 100,000 updates in an environment that rewarded all one and two input logical functions except EQU. Starting at update 100,000, the NOT function was turned into a deleterious trait (i.e., organisms which performed NOT received a 75% fitness reduction). Making a trait deleterious creates selective pressure for the digital organisms to evolve away from that trait and towards other traits. Every next 20,000 updates, another logical operation was turned into a deleterious trait, until only XOR remained. The populations were then evolved for another 100,000 updates in an environment in which only XOR was beneficial and the other 7 functions (7 lower order functions, except EQU) were deleterious. We identified all priming populations in which the final dominant genotype was exclusively performing XOR at the end of the experiment, and randomly selected 25 of the final dominant genotypes to seed populations which would adapt to novel environments.

We used the 25 priming-population final dominant genotypes to seed 2 experiments, each with two paired treatments. Each treatment had 8 populations founded from each priming genotype, for a total of 200 replicate populations per treatment. In the first experiment, populations were adapted to environments with one of two new functions, NOR or EQU, but no other functions. In the second experiment populations were adapted to environments that rewarded XOR and one additional task, NOR or EQU. The new functions were respectively equally complex and more complex than XOR. Environments which rewarded one function were considered to generate a single-peaked fitness landscape and those rewarding multiple functions were considered to generate a multi-peaked landscape. None of the environments punished organisms for performing other logical functions.

All replicates were subjected to two treatments, *Control* and *Replace Deleterious* (RpD). RpD monitored every mutation that arose in the evolving organisms and replaced every deleterious mutation that occurred with a new, randomly chosen neutral, beneficial, or lethal mutation. The RpD protocol is identical to the one used in Covert (2010).

The Control treatment and the RpD treatment differ in that organisms can enter fitness valleys under the Control treatment but not under the RpD treatment. To enter a fitness valley, an organism has to suffer a deleterious mutation. These

mutations are eliminated under RpD. To maintain a comparable overall mutation rate under this treatment, deleterious mutations are replaced by either a neutral or beneficial mutation, which cannot lead into a fitness valley by definition, or by a lethal mutation, which simply kills the offspring organism and prevents any further adaptation along this lineage.

Adaptations in multi-trait environments. We started 50 replicates with a standard seed organism, as before. The priming environment rewarded four traits (NOT, ORN, OR, and NOR). Populations evolved in this environment for 50,000 updates. All evolved populations were then transferred into a novel environment that rewarded all 9 possible one- and two-input logic functions (Table 1). Populations evolved a further 200,000 updates under the novel environment, exposed to two separate treatments: Control and RpD, as before.

Statistical analyses. We carried out all statistical analyses with SciPy (version 0.9). Fitnesses of evolved populations were measured on the dominant (most abundant) genotype in the final population. Fitness comparisons were performed using paired *t*-tests on log-transformed fitness values. When multiple replicates were derived from identical priming populations, we averaged log-transformed fitness values of those replicates before performing paired *t*-tests, to avoid pseudo-replication.

Function Name	Logic Operation	Energy Bonus
NOT	$\sim A; \sim B$	x2
NAND	$\sim(A \text{ AND } B)$	x2
AND	$A \text{ AND } B$	x4
OR_N	$(A \text{ OR } \sim B)$ $(\sim A \text{ OR } B)$	x4
OR	$A \text{ OR } B$	x8
AND_N	$(A \text{ AND } \sim B)$ $(\sim A \text{ AND } B)$	x8
NOR	$\sim A \text{ AND } \sim B$	x16
XOR	$(A \text{ AND } \sim B) \text{ OR } (\sim A \text{ AND } B)$	x16
EQU	$(A \text{ AND } B) \text{ OR } (\sim A \text{ AND } \sim B)$	x32

Table 1: The standard nine logical functions in the Avida environment and their energy bonus. Digital organisms have only the NAND operation available to them and must construct other logical functions out of NAND operations. The energy bonus for each for each function is equivalent to 2^n , where n is the minimum number of NAND operations needed to complete it. Each logical function corresponds to a phenotypic trait in the environment.

Results

Adaptation to environments that reward one or two traits.

How important are deleterious mutations for adaptation to a new environment? To address this question, we carried out experiments with the following general design: We first let populations of digital organisms adapt to a chosen environment. (We call this initial environment the *priming environment* and populations evolved in this environment the *priming populations*.) From the priming populations, we selected a subset that had acquired the optimal phenotype in the priming environment. We then further adapted these populations to novel environments, with two separate treatments. The *Control* treatment was standard adaptation as used for the priming populations. The *Replace Deleterious* (RpD) treatment prevented organisms from experiencing deleterious mutations and hence from exploring fitness valleys. After adaptation in the novel environment, we compared which treatment, if any, led to higher fitness values in the final dominant (most abundant) organisms in the evolved populations.

We first studied the case where the priming environment rewarded one phenotypic trait and the novel environment rewarded either one other or one additional phenotypic trait. For digital organisms in the Avida world, phenotypic traits are defined via two-input logical functions. In our first set of priming adaptations, we evolved organisms to carry out only the XOR function. (We refer to this environment as the *XOR environment*.) We subjected organisms primed in XOR to three novel environments, which rewarded either the NOR function (*NOR environment*), both the NOR and the XOR functions (*NOR/XOR environment*), or both the XOR and EQU functions (*EQU/XOR environment*).

We found that the Control and RpD treatments performed similarly when switching from the XOR environment to either the NOR or the NOR/XOR environment. In NOR, the final dominant fitness was not significantly different among the two treatments (mean pairwise difference of log fitness $\bar{d}=0.83$, $p=0.41$, paired t -test), and neither was the number of times each organism evolved NOR (198 for the control and 196 for RpD). Likewise, in NOR/XOR, the final dominant fitness was not significantly different among the two treatments ($\bar{d}=0.98$, $p=0.34$, paired t -test). The number of times that XOR and NOR evolved in each treatment also did

not differ significantly (NOR 181/185, XOR 181/180 between C/RpD). However, note that in all cases the final fitness under the Control treatment was higher than the final fitness under the RpD treatment, as evidenced by $\bar{d}>0$.

We consider XOR and NOR to be equally difficult to evolve or perform because they require the same number of NAND operations to complete. Thus, the switch from the priming XOR environment to either the NOR or the NOR/XOR environment was a switch to a novel environment of comparable complexity. To assess whether environment complexity played a role in experimental outcome, we also adapted Control and RpD treatments to the EQU/XOR environment. The EQU function is more complex than XOR or NOR because even the most parsimonious solution to EQU requires at least one more NAND operation than either XOR or NOR. In EQU, we found that the Control and RpD groups showed significant differences in both final dominant fitness and the number of replicates that evolved the EQU function (Table 2). The final dominant fitness of the Control group (with deleterious mutations) significantly exceeded that of the RpD group (without deleterious mutations) ($\bar{d}=2.70$, $p=0.012$, paired t -test). The Control group also evolved the EQU function and retained the XOR function significantly more often than the RpD group did (15 more evolved EQU, 20 more evolved both, $p=0.015$ odds ratio 2.57, and $p=0.029$ odds ratio 1.7 Fisher's exact test, respectively).

In summary, these results show that deleterious mutations may be of benefit in long-term adaptation when organisms adapt to a novel environment of increased complexity (EQU/XOR) but not necessarily when they adapt to a novel environment of comparable complexity (NOR, NOR/XOR).

Treatments	XOR	EQU	Both
C	178	189*	159*
RpD	165	170	139

Table 2: Number of replicates which evolved to the tasks present in the XOR/EQU environment. The control evolved EQU and both significantly more than the RpD treatment did ($p=0.015$ and $p=0.029$ respectively, Fishers exact test)

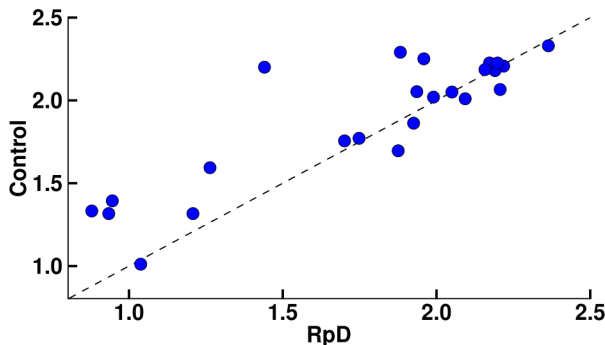


Figure 1: Average \log_{10} final dominant fitness of all replicates initialized with one of 25 seed organism from a priming population, in the XOR/EQU environment. The control treatment tended to have significantly higher fitness than the RpD treatment ($\bar{d}=2.65$, $p=0.014$, paired t -test).

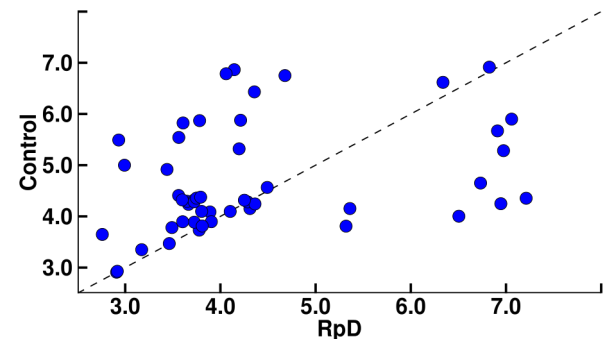


Figure 2: Average \log_{10} final dominant fitness of all 50 replicates that underwent an environmental change from 4 tasks to 9 tasks after 50,000 updates of evolution. No significant differences were detected ($\bar{d}=1.62$, $p=0.11$, paired t -test).

Adaptation to environments that reward multiple traits.

The previous experiments added only a single novel trait. One could hypothesize that if the novel environment rewards many new traits, then beneficial mutations should be plentiful and deleterious mutations should not be required for successful long-term adaptation. On the flip side, a complex novel environment may also provide many fitness barriers which would require deleterious mutations for efficient traversal. To test which of these two scenarios applied in our system, we carried out additional experiments in environments rewarding multiple traits. As priming environment, we used an environment rewarding four traits. As novel environment, we used an environment rewarding all 9 possible one- and two-input logic functions available in Avida. Unlike the previous experiment, we did not select a subset of priming populations that had evolved specific phenotypic traits. We simply evolved 50 replicates for 50,000 updates in the priming environment. We then took all evolved populations and subjected them for another 200,000 updates to the novel environment, under both the Control and RpD treatments as before.

We found that adaptation to the new environment lead to significantly higher fitness values under the control treatment ($d=2.52$, $p=0.0124$, paired t -test). Deleterious mutations seemed to play a key role in the evolution of these populations, despite the introduction of a large number of new beneficial mutations. These new beneficial mutations were presumably on rugged parts of the fitness landscape and inaccessible from the fitness peaks in the earlier environment. The complex logical functions AND and EQU evolved more often under the Control treatment, but not significantly so (see Table 3 for full analysis). These findings suggest that the role of deleterious mutations in changing environments depends not just on the influx of new beneficial adaptations but also on the complexity of the environmental change. Adaptation to more complex environments may require more deleterious mutations.

	NOT	NAND	AND	OR_N	OR
C	196	187	136	198	197
RpD	194	181	113	200	196

	AND_N	NOR	XOR	EQU
C	185	197	51	72
RpD	180	185	46	54

Table 3: Number of replicates that evolved the logical tasks in the multiple traits environment. No significant differences were detected between the control and RpD treatments, but differences were noticeable between the AND and EQU tasks ($p=0.023$ and $p=0.067$ respectively, fishers exact test). However, neither p-value is significant when we consider that the analysis consists of 9 repeated tests, significance would have required a p-value of less than 0.0055.

Discussion

We have shown that deleterious mutations can play a significant role in adaptive evolution, depending on both the initial environment in which a population has evolved and the novel environment to which the organism becomes exposed. We used Avida to evolve digital organisms with and without deleterious mutations in a variety of environments. We found that in a switch from a one-trait environment to a one- or two-trait environment with similar complexity, deleterious mutations did not provide a significant long-term adaptive advantage. However, when we switched to a two-trait environment of greater complexity, we found a significant effect. Likewise, when we switched from a 4 trait environment to a nine trait environment we found a significant role for deleterious mutations. Thus, the effect of deleterious mutations does not seem to be universal; what works in one environment may not work in another. The question then becomes which specific elements of the environment make deleterious mutations advantageous or not. Whitlock (1997) suggested that the type of change may be less important than the frequency of change. Here, we did not assess the impact of frequency of change, since all population experienced exactly one switch from priming environment to novel environment. Thus, our experiments do not speak directly to Whitlock's conjecture. However, our experiments clearly show that frequency of change is not the only relevant variable; changes in environmental complexity and in similarity between the priming and novel environments are sufficient to significantly alter the importance of deleterious mutations to long-term adaptation.

When a population is exposed to a new environments, two parameters should determine how important the traversing of fitness valleys is for successful long-term adaptation: the ruggedness of the fitness landscape in the new environment and the number of novel adaptive opportunities (i.e., number of paths that lead uphill in a fitness landscape). When we switched the environment from *XOR* to *NOR*, *XOR* to *EQU* or to *XOR/NOR*, we likely did not add a large number of novel adaptive opportunities, but we also did not increase the ruggedness of the landscape by much. Thus, traversals of fitness valleys were not particularly important. Switching from *XOR* to *EQU/XOR* similarly did not provide a large number of novel adaptive opportunities, but it likely did increase the ruggedness of the landscape, owing to *EQU*'s increased computational complexity relative to *XOR*. Thus, traversals of fitness valleys were important in this scenario. Finally, by switching from the four-trait environment to the nine-trait environment, we introduced additional ruggedness, but we certainly also added a large number of additional adaptive opportunities. Many of the new adaptive opportunities seem to require a deleterious mutation in order to exploit them, as evidenced from the higher fitness values in the control treatment.

Fitness interactions between genes can sometimes create fitness effects that deviate from our expectations. This effect is called epistasis, and can occur when two or more mutations have a greater or smaller sum fitness together than they do individually. Sign-epistasis is the most extreme form of this, in which individually deleterious mutations may provide a net benefit (or vice-versa). Recent theoretical works

have suggested that while escape from fitness peaks via sign-epistatic mutations is possible (Weinreich and Chao 2005, Weissman et al. 2010), the main role of sign-epistasis is to constrain adaptive paths. Our work suggests that sign-epistasis may be a driving force in evolution.

One limitation of our work is that we did not identify the key adaptations that caused the net fitness benefit. Previous works with deleterious mutations in digital organisms identified and isolate key sign-epistatic interactions on the line of descent that opened up new areas of the adaptive landscape to explore (Covert 2010). A second limitation is that the seed organisms for all experiments were evolved in populations that allowed deleterious mutations. It is possible that the initial exposure to deleterious mutations made it easier for adaptation in later experiments to proceed without additional deleterious mutations.

Evolution is a combination of three factors: variation, inheritance, and selection. Most works on evolution focus on just one of these factors: selection. Variation and inheritance may be thought of as synonyms for chance and history, without which evolution cannot proceed (Blount et al 2010). There is no better example of a chance evolutionary event becoming important than a sign-epistatic fitness reversal, a previously deleterious mutation which becomes critically important to future evolution. With modern technology it is now possible to observe and measure the importance of history and chance in simple evolving systems. While selection is the ultimate arbiter of which variation and which history will be successful, selection requires raw material to work with, and this raw material may be the result of highly unlikely events.

Acknowledgments

This material is based in part upon work supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454 and by the Freshman Research Initiative at the University of Texas at Austin. We thank Charles Ofria and Richard Lenski for stimulating discussions.

References

- Blount, ZD., Borland CZ., and Lenski, RE (2008), Historical Contingency and the Evolution of a Key Innovation in an Experimental Population of *Escherichia coli*. *PNAS* 105:7899-7906.
- Covert III, A. W. (2010), *On the Beneficial Effects of Deleterious Mutations*. Ph.D thesis, Department of Computer Science Michigan State University.
- Cowperthwaite MC, Bull JJ, Meyers LA (2006), From bad to good: Fitness reversals and the ascent of deleterious mutations. *PLoS Comp Biol* 2:1292-1300.
- Kvitek DJ, Sherlock G (2011), Reciprocal Sign Epistasis between Frequently Experimentally Evolved Adaptive Mutations Causes a Rugged Fitness Landscape. *PLoS Genet* 7:e1002056.
- Lenski RE, Ofria C, Collier TC, Adami C (1999), Genome complexity, robustness and genetic interactions in digital organisms. *Nature* 400:661-664.
- Lenski RE, Ofria C, Pennock RT, Adami C (2003), The evolutionary origin of complex features. *Nature* 423:139-144.
- Ofria C, Wilke CO (2004), Avida: a software platform for research in computational evolutionary biology. *J Artif Life* 10:191-229.

- van Nimwegen E, Crutchfield JP (2000), Metastable evolutionary dynamics: Crossing fitness barriers or escaping via neutral paths? *Bull Math Biol* 62:799-848.
- Weinreich DM, Chao L (2005), Rapid evolutionary escape by large populations from local fitness peaks is likely in nature. *Evolution* 59:1175-1182.
- Weissman DB, Desai MM, Fisher DS, Feldman MW (2009) *J Theor Biol* 258:286-300.
- Whitlock, MC. (1997), Founder Effects and Peak Shifts Without Genetic Drift: Adaptive Peak Shifts Occur Easily When Environments Fluctuate Slightly. *Evolution*. 51:1044-1048
- Whitlock, MC., Phillips PC., Moore, FBG., and Tonsor, SJ. (1995), Multiple Fitness Peaks and Epistasis, *Annu. Rev. Ecol. Syst.* 26:601-629

What does sex have to do with it: tracking the fate of deleterious mutations in sexual populations.

Arthur W. Covert III^{1,2*}, Lane Smith^{1,2}, Dakota Z. Derrberry^{1,2}, and Claus O. Wilke^{1,2}

¹ Section of Integrative Biology, Center for Computational Biology and Bioinformatics, and Institute for Cellular and Molecular Biology, The University of Texas at Austin.

² BEACON, Center for the Study of Evolution in Action,

*Author to whom correspondence should be sent

Art.Covert@austin.utexas.edu

Abstract

Deleterious mutations sometimes revert into beneficial mutations via epistatic interactions with subsequent mutations. This type of interaction among mutations is called “sign-epistasis.” Recent works have explored the role of sign-epistasis in the evolution of asexual populations. Some have indicated that the fixation of sign-epistatic deleterious mutations may be critical for adaptive evolution. However, sign-epistasis is considered to be important only for asexual populations, because recombination in sexual populations tends to disrupts linkage between epistatically interacting mutations. Here, we tested the hypothesis that recombination prevents adaptation via sign-epistatic fitness reversions, by examining deleterious mutations in sexually-reproducing digital organisms. We examined every deleterious mutation that arose on the genealogy between the original ancestor and the final dominant genotype (the “graph of descent”). We show that sign-epistatic pairs of mutations emerged in several replicate populations, and that they contributed positively to the long-term adaptation of the population.

Introduction

Deleterious mutations are generally thought of as a drag on adaptive ability. In rare cases a deleterious mutation may be joined by a compensatory adaptation that ameliorates the deleterious effect. If the compensatory adaptation itself was deleterious in the absence of the original deleterious mutation, then the pair of mutations are individually deleterious, but jointly beneficial. In finite populations these pairs of mutations may then sweep to fixation jointly, rather than having to fix sequentially (Iwasa et al 2004, Weissman et al 2009).

Interactions between mutations that alter their cumulative fitness effect are called epistatic interactions. The most extreme form of epistasis, a change in the fitness effect from deleterious to beneficial, is called sign-epistasis (Weinreich and Chao, 2005).

Theoretical works examining sign-epistasis have found that deleterious mutations in asexual populations can segregate at low frequencies and occasionally be compensated by subsequent mutations. Computational simulations have found actual examples of sign-epistasis contributing to adaptation (Lenski et al 2003, Cowperthwaite et al 2006, Covert 2010). More recently, an example of a sign-epistatic interaction has

been found in *Saccharomyces cerevisiae*, the first known discovery of a sign-epistatic fitness reversal in an organic system (Kvitek and Sherlock 2011).

All works on sign-epistasis to date have considered asexual systems. Presumably, recombination would disrupt any useful sign-epistatic pairing, unless the interacting mutations were very tightly linked (Weinreich and Chao 2005). We test this assumption using digital organisms that undergo recombination. We compare two experimental treatments, one in which organisms can suffer deleterious mutations and one in which deleterious mutations are prevented from occurring. We see that populations that do not experience deleterious mutations evolve to significantly lower fitness than populations that do. To identify where sign-epistatic mutations occurred and what effect they had, we reconstruct a complete genealogy (“graph of descent”) from the original ancestor of the population to the final, most abundant genotype. We use the graph of descent to isolate specific examples sign-epistatic interactions and examine when they emerged, when they recovered, and what magnitude their epistatic effects had.

Methods

Experimental system. We used the digital-life platform Avida (version 2.12.2) for all experiments. The Avida world holds a population of digital organisms. Digital organisms are self-replicating and evolving computer programs, written in a special-purpose programming language and executed on a grid of virtual CPUs. The computer program defining a digital organism is considered to be the organism's genome. Mutations are random changes in the genome. Here, we only used point mutations, which replace one instruction in the genome with a randomly chosen instruction.

Digital organisms are rewarded with additional energy (CPU time) for the successful computation of logical functions. Thus, there is a selective pressure for digital organisms to evolve the capability to efficiently compute multiple logical functions. Here, we used the standard Avida “logic-9” environment as described in (Lenski et al. 1999, Lenski et al. 2003). This environment rewards one- and two-input logical functions; reward amounts increase with difficulty of the logical function to compute.

Adaptation experiment. We adapted replicate populations of digital organisms for 250,000 updates¹. Populations were seeded with an organism whose genome consisted of 50 instructions. The seed organism could self-replicate but not perform any of the logical functions. Populations were seeded with a single digital organism; population size rapidly grew to a maximum carrying capacity of 10,000, at which it remained for the remainder of the adaptations. Organisms had a 25% chance of experiencing a single point mutation on divide (replication). This mutation rate translates into a 0.5% probability of mutation per site per generation.

Recombination was implemented as follows: When an organism divided, its offspring was placed into a birth chamber. Organisms remained in the birth chamber until they were joined by another organism, with which they recombined. Recombination occurred at a single cross-over point in the middle of the genome. Recombination between identical genotypes was allowed (Misevic et al 2006).

We ran 50 replicates each of two separate treatments, *Control* and *Revert Deleterious* (RvD). In the Control treatment, all newly divided organisms were replaced in an isolated test environment before they entered the birth chamber. Each organism was tested to determine if it could self-replicate without altering its genome. Those organisms that could self-replicate stably were placed in the birth chamber. Organisms that could not self-replicate stably were sterilized and removed from the population.

In the RvD treatment, we tested organisms for stable self-replication as well as for the presence of a deleterious mutation. If an organism experienced a deleterious (but not lethal) mutation, we reverted the organism's genotype to the parent's genotype². The RvD treatment prevented the occurrence of deleterious point mutations. However, note that recombination could nevertheless create combinations of mutations that were deleterious, even if the individual mutations were not deleterious in their parent organisms.

We used a structured population, to make constructing the graph of descent more tractable. Each population was divided into 100 subpopulations of equal size. Normally when an organism leaves the birth chamber it is placed next to the last parent to contribute a genome to the birth chamber. When organisms left the birth chamber in our structured environment they had a 1 in 20,000 chance of migrating to a new subpopulation (approximately one migration event every other generation). The structured population limits the total number of genomes that may contribute to the final population. Since Avida saves only those genotypes related to the most abundant genotypes still alive in the population, the choice of a structured population made both the file size and the number of genotypes to examine more manageable.

For each evolved population, we measured the fitness of the dominant (most abundant) genotype at the end of 250,000 updates of adaptation. We used this fitness measure to assess long-term evolutionary success of evolved populations.

Construction of the graph of descent. We reconstructed the genealogies of 20 final dominant genotypes (FDGs) from the

first 20 replicates in the experiment. Each genealogy included all parent-offspring relationships between the original ancestor and the FDG. We began by identifying the two parents of the FDG. We next identified the parents of the FDG's two parents. Then we identified the parents of the FDG's parent's parents. We continued until we had traced back the entire ancestry to the original ancestor. This procedure resulted in a graph of genotypes spanning from the original common ancestor to the FDG, with bidirectional edges from parent to offspring genotypes.

We modified the Avida software to output the mutations and crossover points for each genotype. Starting with the offspring of the common ancestor, we examined the fitness effect of every mutation on the graph of descent. For each deleterious mutation on the graph of descent we created a "mutation subgraph" that traced the fate of a single mutation. Each deleterious mutation was tracked from its entrance in the graph until either all genotypes on the graph had mutated away from the deleterious mutation or one of the genotypes had undergone a sign-epistatic fitness recovery.

Identification of sign-epistatic mutations on the graph of descent. We used the following algorithm to identify sign-epistatic mutations: We iterate over all deleterious mutations on the graph of descent. For each mutation, we confirm its deleterious fitness effect by undoing only the mutation, not the recombination. We undo or revert a mutation by replacing the mutated instruction with the instruction at the same locus in the parent of the origin. If the fitness of the reverted offspring is less than the fitness of at least one of the parents, we know the mutation is deleterious. We call the first genotype that contains a deleterious mutation the origin (Figure 1). For each offspring containing the deleterious mutation, we similarly revert the mutation to the instruction at the same locus in the parent of the origin. If the reversion increases fitness we know that the mutation is still deleterious in the current genetic background. If the reversion decreases fitness on the current background (or does not alter it), we know that the previously deleterious mutation has undergone a sign-epistatic fitness reversal and is now beneficial.

We continue down the subgraph, checking the fitness effect of the deleterious mutation in all descendants. We first check all genotypes of equal depth, d , from the origin before checking those at depth $d+1$ (Figure 1). If we find a descendant that does not contain the deleterious mutation we prune that descendant's children from the mutation subgraph. Eventually we will reach one of two outcomes: (1) the deleterious mutation has undergone a fitness reversal and is no longer harmful to the descendant or (2) no more descendants contain the deleterious mutation. In the second outcome, the original deleterious mutation was purged from the environment before it underwent a sign-epistatic change. In the first outcome, the original deleterious mutation became beneficial through interaction with a subsequent mutation..

Results

Deleterious Mutations contribute to long-term adaptive success in sexual populations. That occasional deleterious mutations can contribute to long-term adaptive success in asexual populations has been observed in a variety of studies (Lenski 2003, Cowperthwaite et al 2006, Covert 2010), but

¹ A unit of time in avida equal to 30 instruction executions per living organism in the population, see Ofria and Wilke (2004) for further details.

² Each offspring can experience at most one mutation, by design.

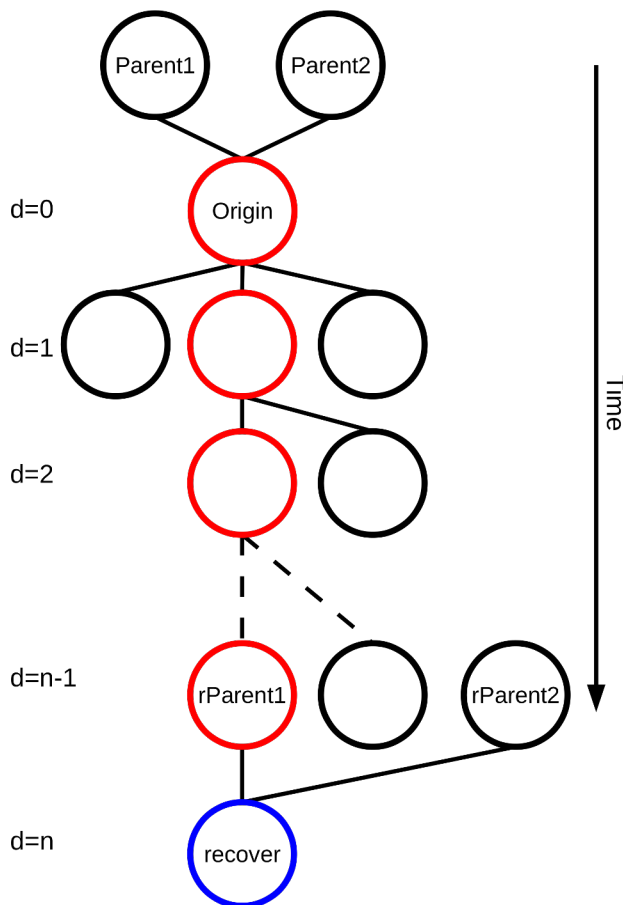


Figure 1: Tracking a deleterious mutation through the graph of descent. We mark the genotype that contains the first instance of the deleterious mutation as the “origin.” The deleterious mutation that emerges in the origin genome is traced through time. At each step, the mutation’s fitness effect is tested. When a fitness reversal occurs, we mark the genotype it occurs in as “recover.”

not in sexual populations. To measure the impact of deleterious mutations in sexual populations, we evolved replicate sexual populations of digital organisms under two treatments, *Control* and *RvD* (revert deleterious). The *Control* treatment consisted of standard adaptation. The *RvD* treatment was identical to *Control*, with the exception that we monitored all mutations in offspring organisms (after division but before recombination) and determined whether an offspring organism had suffered from a deleterious (but not lethal) mutation. We reverted those offspring organisms with a deleterious mutation to the parental genotype. After reversion of a deleterious mutations, offspring organisms were subjected to recombination with other offspring organisms, as in the *Control* treatment.

We adapted 50 replicate populations under both treatments. We found that the dominant genotypes after adaptation had, on average, significantly higher fitness in the *Control* treatment than in the *RvD* treatment (Figure 2, $p=1.74 \times 10^{-4}$, $h=705.0$, U-test). This finding strongly suggests that deleterious mutations contributed to the long-term evolutionary success of the *Control* populations. How exactly

the deleterious mutations impacted adaptation remains unclear.

Classification of Individual Deleterious Mutations. How do deleterious mutations benefit long-term adaptation? We hypothesized that a fraction of deleterious mutations underwent a sign-epistatic fitness reversal. We developed an efficient algorithm to track the graph of descent in an asexual population and to test for the presence of sign-epistatic mutations on this graph. The algorithm is described in detail in the methods. In brief, the algorithm works as follows: Every mutation that lowers fitness is examined in every genotype that carries it on the graph of descent. When the mutation no longer harms its current genotype we know that it has undergone a sign-epistatic fitness-reversal. In Figure 1, the first genome that expresses the deleterious mutation is called the “origin.” The “recovery” is the genotype that carries the mutation, but is the first instance where the mutation is no longer deleterious. The depth of recovery is the number of steps between the origin genotype and the recovery genotype.

Characteristics of sign-epistatic mutations in sexual populations. We ran our analysis of individual deleterious mutations on the first 20 replicates. Among the 6,921,517 analyzed genotypes, we found a total of 22,724 deleterious mutations. We limited our analysis to mutations that caused a fitness loss of over 1% relative to both parents, and whose effects were fully reversed via sign-epistasis. We found 902 such mutations.

Figure 3 displays the fitness cost of the 902 deleterious mutations versus their depth of recovery. Fitness cost was measured relative to the average fitness of both parents. The depth of recovery is the number of steps on the mutation subgraph from the origin of the deleterious mutation (see Figure 1). The average depth of recovery was fairly small,

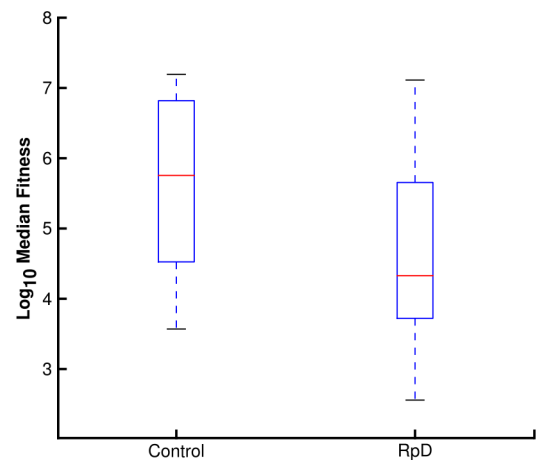


Figure 2: Effect of deleterious mutations on final organism fitness. 50 replicate populations were evolved with (control) and without (RpD) deleterious mutations. Control populations had significantly higher fitness ($p=1.74 \times 10^{-4}$, $h=705.0$, U-test) than the RpD population. The fitness differential suggests that deleterious mutations contributed positively to adaptive evolution, most likely via fitness-reversals.

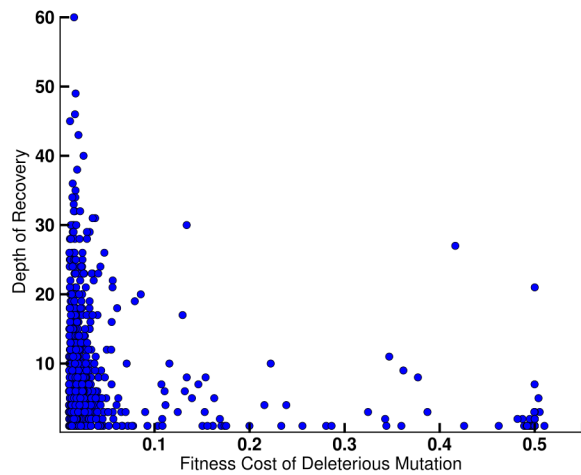


Figure 3: Fitness cost of 848 deleterious mutations (that underwent a fitness reversal) versus their depth of recovery. The x-axis shows the initial percentage of fitness loss relative to the average fitness of both parents. The y-axis shows the number of steps d on the graph of descent between the origin of the deleterious mutation and its recovery (see Figure 1). Most recoveries (771) occur a short distance from the origin and rescue mutations with relatively modest initial fitness cost (less than 10%). An additional 54 mutations had a fitness loss greater than 55% off fitness and are not shown.

7.80 steps on the graph of descent, although the standard deviation of depth of recovery was large, 7.93. Approximately 91.3% of recoveries took less than 20 steps, although one took 60. The average fitness cost of a deleterious mutation was 10.0% of average parent fitness, and the standard deviation was high, 22.4%. While a few deleterious mutations on the graph of descent had some extreme fitness losses (approximately 61 occurred between 50% and 99.9% fitness loss), most of the fitness losses were modest but would have been harmful had they not been compensated for.

Figure 4 shows the percent increase in fitness on recovery (i.e., the amount of fitness increase relative to the parents on the origin) versus the depth of recovery. The high increase in fitness at the time of recovery strongly suggests that the deleterious mutations contributed to the evolution of logical functions rewarded by the environment. The average increase in fitness was approximately 4.84%. The standard deviation of fitness increase was large, 434%. The smallest fitness gain was 1% while the largest fitness gain was over 3,000%. Such large fitness gains are normally only associated with the evolution of more complex logical operations (Lenski et al 2003, Covert 2010). Therefore, some deleterious mutations were likely instrumental in the evolution of complex features. The vast majority of fitness increases (747) were between 1% and 10%, suggesting optimizations of the genotypes replication efficiency.

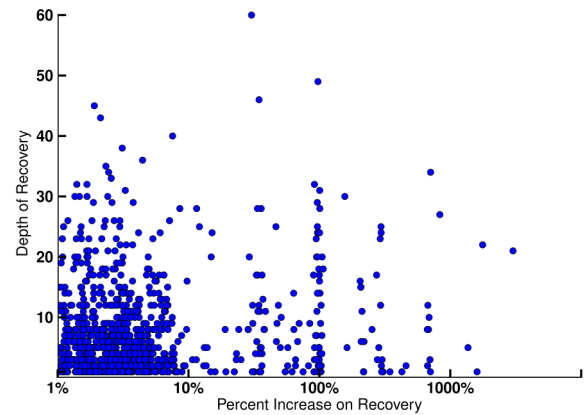


Figure 4: Percent fitness increase relative to the parent plotted against the depth of recovery. The x-axis shows the fitness increase of the recovery genotype relative to the parents of the origin (see figure 1). The y-axis shows the number of steps d on the subgraph between the origin of the deleterious mutation and its recovery. Most recoveries (747) are correlated with a modest but important fitness increase between 1% and 10%.

Discussion

In large asexual populations sign-epistatic mutations may appear sequentially and sweep to fixation together (Weinreich and Chao 2005, Weissman et al. 2009). Populations that experience a sweep of sign-epistatic pairs of mutations may be able to pass through fitness valleys. In sexual populations sign-epistatic mutations may be brought together by recombination, but will be disrupted by recombination unless they are tightly linked.

Analytical works have shown that recombination at low levels does not disrupt the fixation of sign-epistatic pairs, but that linkage-disequilibrium takes over at higher levels of recombination (Weinreich and Chao 2005). This implies that there is a critical recombination rate beyond which the simultaneous fixation of sign-epistatic mutations is highly improbable. Our initial experiments suggest that fitness-reversals play an important role, despite a high recombination rate. While there are many other factors which must be accounted for, it seems clear that the number and frequency of sign-epistatic events is less important than where they carry the population on the fitness landscape.

We found that deleterious mutations may contribute positively to the long-term adaptation of sexual populations. Eliminating deleterious mutations from evolving populations significantly decreased the fitness of the final dominant genotypes in replicate populations. We have demonstrated that it is possible to track the entire graph of descent in sexual populations of digital organisms, from the initial ancestor to the final dominant genotype, and to track epistatic interactions among mutations on the graph. We found numerous examples of sign-epistatic recoveries in populations that experienced deleterious mutations. In these examples, mutations caused a fitness loss in the genetic backgrounds in which they arose,

but interacted epistatically with subsequent mutations and eventually contributed positively to fitness.

Weissman DB, Desai MM, Fisher DS, Feldman MW (2009) *J Theor Biol* 75:286-300.

Our sexual populations were strictly haploid, the introduction of diploidy or polyploidy could create the added complexity of considering dominant and recessive traits. In addition, we had only a single crossover point. The effect we observed may be disrupted as additional crossover points increase linkage-disequilibrium between traits. However, previous work in digital organisms has shown that recombination encourages organisms to evolve more modular genotypes (Misevic et al. 2006). Therefore, additional crossover points may also encourage mutations effecting a single trait to be more tightly linked on the genome. Further work is necessary to resolve this issue.

Our work indicates that there are circumstances when sign-epistasis may play an important role in the evolution of sexual populations. It opens new possibilities for researching the exploration of fitness landscapes in sexual populations. Asexual studies of fitness landscapes generally assume that populations will move from one genotype to a genotype separated by only a few point mutations. In sexual systems, populations may move in great leaps and bounds across the fitness landscape, due to recombination. Our graph of descent gives us the ability to observe, for the first time, the movement of sexual populations through fitness landscapes.

Acknowledgments

This material is based in part upon work supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454 and by The Freshman Research Initiative at the University of Texas at Austin. The authors acknowledge Charles Ofria and Richard Lenski for valuable conversations.

References

- Covert III, A. W. (2010), *On the Beneficial Effects of Deleterious Mutations*. Ph.D thesis, Department of Computer Science Michigan State University.
- Cowperthwaite MC, Bull JJ, Meyers LA (2006), From bad to good: Fitness reversals and the ascent of deleterious mutations. *PLoS Comp Biol* 2:1292-1300.
- Iwasa Y, Michor F, Nowak MA (2004) Stochastic tunnels in evolutionary dynamics. *Genetics* 166:1571-1579.
- Kvitek DJ, Sherlock G (2011), Reciprocal Sign Epistasis between Frequently Experimentally Evolved Adaptive Mutations Causes a Rugged Fitness Landscape. *PLoS Genet* 7:e1002056.
- Lenski RE, Ofria C, Collier TC, Adami C (1999), Genome complexity, robustness and genetic interactions in digital organisms. *Nature* 400:661-664.
- Lenski RE, Ofria C, Pennock RT, Adami C (2003), The evolutionary origin of complex features. *Nature* 423:139-144.
- Misevic D, Ofria CA, Lenski RE (2006) Sexual reproduction reshapes the genetic architecture of digital organisms. *Proc R Soc London B* 273:457-464.
- Ofria C, Wilke CO (2004), Avida: a software platform for research in computational evolutionary biology. *J Artif Life* 10:191-229.
- van Nimwegen E, Crutchfield JP (2000), Metastable evolutionary dynamics: Crossing fitness barriers or escaping via neutral paths? *Bull Math Biol* 62:799-848.
- Weinreich DM, Chao L (2005), Rapid evolutionary escape by large populations from local fitness peaks is likely in nature. *Evolution* 59:1175-1182.

Heterochronous Neural Baldwinism

Keith L. Downing

Department of Computer Science
The Norwegian University of Science and Technology
Trondheim, Norway
keithd@idi.ntnu.no

Abstract

Neural Baldwinism concerns the Baldwin Effect in the evolution of brains and intelligence. The first phase of the Baldwin Effect (B.E.), wherein plasticity provides a selective advantage, is intuitive and commonplace in simulations of adaptive systems. However, the second (assimilation) phase often poses problems for Baldwinism in general, and this is particularly acute for biological neural networks, where a complex developmental process greatly confounds the mapping from genotype to functional phenotype: a brain whose synapses are tuned to perform particular tasks. Since a strong genotype-phenotype correlation is often viewed as a prerequisite to this second phase, the body's most plastic organ would appear to defy Baldwinism. However, a detailed examination of 3 key processes of neural adaptation blurs the distinction between classic developmental and learning stages of brain maturation, thus supporting a re-interpretation of Neural Baldwinism's phase II as a heterochronous shift of the bulk of these three adaptive processes from postnatal to prenatal stages. This article illustrates Heterochronous Neural Baldwinism (HNB) with artificial neural networks that evolve, develop and learn, and in which some degree of synaptic tuning shifts to the prenatal stage.

Introduction

The Baldwin Effect (B.E.) (Baldwin, 1896; Turney et al., 1997) concerns the ability of learning to accelerate evolution via a two-stage process. In phase I, individuals with phenotypic plasticity achieve higher fitness than those relying purely on innate skills. This moves the population distribution toward plastic individuals. In phase II, some of these learned skills become innate by chance mutations. This assimilation of plastic features into the genome and developmental process becomes more probable when the genotype-phenotype mapping is not overly complex (with correlations maintained between genotype and phenotype spaces); and selection pressure favors assimilation when a) the environment is reasonably static across the generations, and b) learning has a fitness cost (Mayley, 1996).

Although B.E. seems plausible for some phenotypic traits, such as the size of muscles and the efficacy of certain physical skills, its relationship to the evolution of intelligence is more tenuous, given contemporary understanding of the

brain, neural development and synaptic change. If learning is generally equated with synaptic change, then how can the modification of a few of the (human) brain's 100 trillion synapses be assimilated into DNA consisting of approximately 25,000 genes? In general, the mapping from genotype to phenotype is highly *indirect*, and correlations between genotype space and neural network space seem highly unlikely, once again precluding the assimilation of specific synaptic change into the genome.

In search of a more plausible reconciliation of neural evolution and Baldwinism, we examine the mechanisms traditionally associated with neural-network development and learning; the border between the two seems fuzzy with respect to the creation of new neurons (a.k.a. neurogenesis) and synapses (a.k.a. synaptogenesis), along with the tuning of those synapses. For instance, neurogenesis and synaptogenesis are not restricted to early neuro-development, as once believed. Recent evidence (Shors, 2009) shows that neurons can be generated and inter-connected throughout life, depending upon an animal's mental (and physical) challenges. Also, a good deal of synaptic tuning has been shown to occur prenatally (Sanes et al., 2006). Thus, neuro- and synaptogenesis, along with synaptic tuning, can be shared between development and learning, with the genome brokering the actual division of labor. Furthermore, heterochronous shifts in these distributions - that transfer some of the burden from stages of life that are strongly influenced by the environment (a.k.a. nurture) to those strongly governed by the genes (a.k.a. nature) - seem to support the assimilatory requirements of B.E. phase II.

Motivated by these biological findings and their implications for the B.E., we investigate models in which a) artificial neural networks (ANNs) evolve, develop and learn, b) the core processes of neurogenesis, synaptogenesis and synaptic tuning have varying levels of activity in early (developmental) and late (learning) stages of life, and c) these levels are determined by the genome. Then, by monitoring the evolving distribution of these 3 processes among development and learning, we observe this more flexible interpretation of Neural Baldwinism.

Related Work

Hinton and Nowlan's simulations - classic in their simplicity and elegance - first illustrated B.E (Hinton and Nowlan, 1987). They showed that early learning helped guide evolution toward a difficult goal (B.E. phase I), but as the population approached the target, the flexible portions of the phenotype became hard-wired to the correct values, thus jettisoning the (costly) learning capabilities (B.E. phase II). Their model involved simple bit-string genotypes, which doubled as phenotypes, so no development nor neural networks were involved, though they commented that the model could serve as a coarse abstraction for the evolution of neural networks.

In another seminal Baldwinian simulation, (Ackley and Littman, 1992) showed the B.E. in evolved pairs of interacting neural networks, one of which learned by back-propagation while the other evolved (but could not learn) to provide proper world-state evaluations to guide learning in the former network. Upon adding learning (in neural networks) to cellular encoding (CE) (Gruau and Whitley, 1993) observed the confounding effects of development (in CE) upon the B.E.. Later, (Downing, 2004) extended the Hinton and Nowlan model to include an abstract developmental process based on a Turing machine (TM) (whose specifications were encoded in the genome). Those experiments showed the scaffolding effect that development can manifest to reduce the learning burden and thus support B.E. phase II. That scaffolding effect is also evident in this article, but now with fully-functioning neural networks as the phenotype and developmental synaptic tuning replacing the TM.

Recently, (Paenke et al., 2009) proposed a mathematical framework to help quantify when, in fact, learning will accelerate evolution, while many important B.E. studies employ models other than neural networks (Suzuki and Arita, 2007; Bull, 1999; Mayley, 1996) to reveal critical relationships between fitness landscapes, epistasis, the genotype-phenotype mapping, and B.E. Of critical relevance to this article are Mayley's two key prerequisites for B.E. phase II: a) a strong correlation between genotype and phenotype space, and b) a significant learning cost.

Despite the obstacles to B.E. posed by neural development, a plausible reconciliation of the two involves a heterochronous shift of a significant degree of neurogenesis and synaptogenesis from postnatal (experience-driven) learning to the prenatal (gene-governed) phase of life, as shown in (Downing, 2010). In this article, we turn to the third key factor, synaptic tuning, and explore the degree to which it too can be assimilated into neural development.

Heterochronous Neural Baldwinism (HNB)

Figure 1 summarizes a few of the key processes involved in the mapping from genes (roughly 25,000 in humans) to the brain (containing around 100 billion neurons and 100 trillion synapses). The high degree of scrambling and elaboration of genetic information that occurs during neural development

and tuning must clearly ruin any correlations between genotype and phenotype space, thus violating a key precondition for B.E. phase II (Mayley, 1996). Thus, a plausible model of Neural Baldwinism would seem to require a different perspective and/or abstraction level.

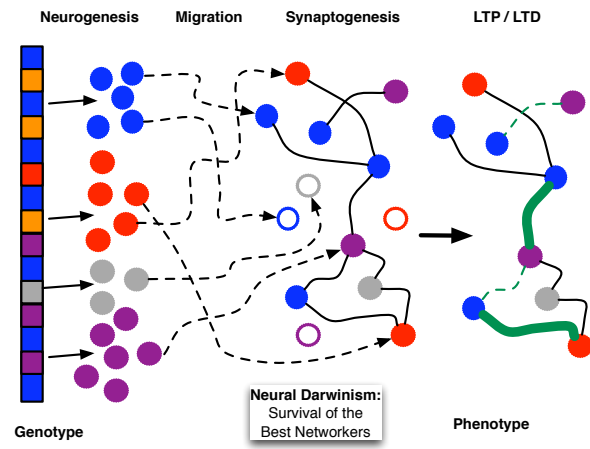


Figure 1: The complex gene-to-brain mapping.

A common *lock-step* scenario for brain formation and maturation consists of two clearly distinct phases: a) prenatal development wherein neurons are produced and linked together, and b) postnatal learning, wherein synaptic strengths are modified to enhance behavioral control. Though convenient for computational models and general explanations, this over-simplifies temporal relationships whose details may prove useful for understanding Neural Baldwinism. For example, many studies, summarized in (Sanes et al., 2006), find high levels of long-term potentiation (LTP) and long-term depression (LTD) - both forms of synaptic tuning - during prenatal development. In fact, the rates of LTP and LTD (i.e. learning rates) are actually very high during development and much lower during adult life. In addition, recent work (Shors, 2009) reveals that a) neurogenesis occurs throughout life, particularly in the dentate gyrus (DG) of the hippocampus, but b) those neurons only hook up to other neurons (and ultimately survive) if the organism subsequently performs cognitively-challenging tasks. Thus, although we can retain terminology that equates development with all prenatal brain formation, and learning with postnatal activity, the constituent processes of development and learning are clearly not mutually exclusive in this (more biologically realistic) *overlapping* model.

This new perspective motivates a reinterpretation of B.E. in neural networks. In a lock-step model, B.E. phase II entails converting synaptic-strength changes (i.e. classic learning) into genomic codes for controlling neurogenesis and synaptogenesis (i.e. classic development). This represents

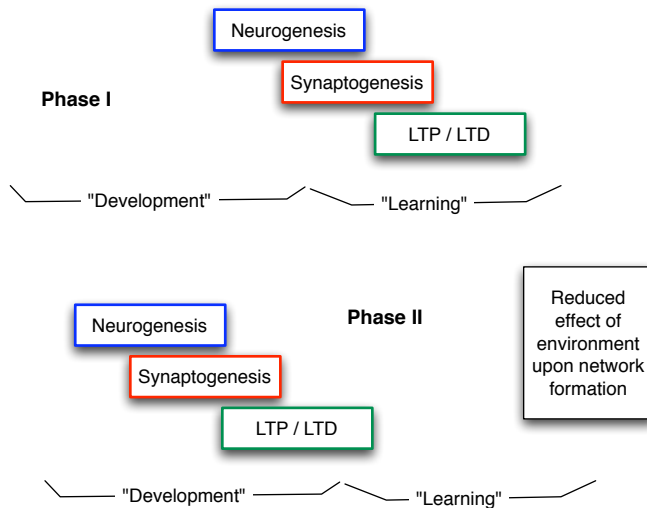


Figure 2: Overview of Heterochronous Neural Baldwinism, wherein the hallmark of phase II is the transfer of considerable neuro- and synaptogenesis, along with synaptic tuning, from postnatal to prenatal stages.

the reverse encoding of the results of one process into two dramatically different processes. However, in an overlapping model, the assimilation phase involves only quantitative, not qualitative, change.

Thus, B.E. Phase II may primarily constitute heterochrony: a change in the *onset*, *termination* and *rates* of neurogenesis, synaptogenesis and LTP/LTD across an organism's life stages, as shown in Figure 2. Under the view that adaptive changes in later life are predominantly governed by the environment, not the genome, a Baldwinian modification could simply be to move more of that adaptive change into earlier life stages, where genomic control may dominate.

For example, when the biochemical bases for LTP and LTD arose in evolution, both processes may have been very active throughout life, requiring constant environmental signaling to tune neural circuitry. However, over many (thousands of) generations, genomic changes could have arisen such that the early stages of development utilized neurogenesis, synaptogenesis and high LTP/LTD to form much of this circuitry with a minimum of environmental influence. Similarly, the rates of neurogenesis and synaptogenesis could have originally been much less variable throughout life, but evolution has gradually found genomes coding for an acceleration of these processes in early development; and thus, more of these activities became governed by genomic rather than environmental factors. The neural plasticity that remains in today's adult genomes (of any species), may represent that flexibility which evolution found optimal with respect to factors such as a) the coding limits of the genome,

b) constraints of the animal's brain and body, and c) earth's environment and the rates of change associated with it.

This research illustrates Heterochronous Neural Baldwinism using ANNs which a) evolve using genetic algorithms, b) learn via backpropagation, and c) employ a complex developmental procedure for tuning synaptic weights prior to backpropagation. Importantly, evolution controls both the extent of backpropagation and the detailed nature of developmental tuning.

In this model, phase II of the Baldwin Effect is evident when the genome transfers a significant level of synaptic tuning from postnatal to prenatal stages, i.e., from a life stage when the environment governs a good deal of neural activity to a stage when the genome holds more control. Hence, although this model employs a complex, correlation-destroying mapping from genes to adaptive (developmental and learning) parameters to synaptic weights, the genome still possesses the ability to evolve the recipe for a developmental procedure that can reduce the burden of postnatal adaptation and thereby increase phenotypic fitness.

Developmental Synaptic Tuning (DST)

The DST model introduces a biologically-inspired mechanism for modifying connection weights prior to exposure to the environment (i.e., training cases). This mechanism abstracts from neurological studies showing that spontaneous waves of neural activity, modulated by cyclic-AMP (cAMP) concentrations, lead to early synaptic tuning during development, prior to the exposure to normal sensory inputs. This has been shown to play an important role in the binocular segregation of connections from the retina to the lateral geniculate nucleus (LGN) (Stellwagen and Shatz, 2002), while others (McNaughton et al., 2006) postulate similar wave-induced synaptic tuning in the hippocampus, and a variety of evidence, summarized in (Sanes et al., 2006), indicates both a) the presence of these waves throughout the brain during neural development, and b) their instructive role in synaptic formation and tuning.

These waves promote neural firing such that neurons in adjacent regions that happen to fire simultaneously (due to stimulation from their respective activation waves) will have their synaptic connections modified, typically by Hebbian means. Thus, early chemical waves strongly influence the patterning of neuronal connections, prior to the molding effects of normal sensory stimuli.

A comprehensive model of this phenomena would include the chemical and physical bases of reaction-diffusion processes, a reasonably straightforward but computationally-intensive endeavor. Fortunately, compositional pattern-producing networks (CPPNs) (Stanley, 2007) provide an efficient alternative for abstractly modelling any number of natural pattern-generating processes.

Composite Pattern-Producing Networks (CPPNs)

As shown on top of Figure 3, a CPPN (Stanley, 2007) resembles a neural network, but with each node housing one of a number of alternative activation functions, as opposed to the standard sigmoids, step functions and hyperbolic tangents of ANN nodes. For example, the CPPN may include Gaussians, absolute values, and sine waves (as well as the common ANN activation functions). Each CPPN connection includes a weight, and all nodes compute the sum of their weighted inputs, which serves as input to the activation function, whose result becomes the node's output.

The CPPNs in this research have no explicit layered organization (other than pre-defined input and output nodes), so any node can send outputs to any other node; and all nodes (except the inputs) can receive weighted outputs. At each timestep, the nodes undergo asynchronous activation, wherein each node simply sums the weighted outputs in its input buffer and feeds that sum to its activation function to produce an output value, which is immediately propagated to the input buffers of all post-synaptic neighbors. After a user-determined number of update rounds, the CPPN's outputs are gathered from the output nodes.

By sending Cartesian coordinates through a CPPN and using the output value to encode pixel color or intensity, the CPPN can generate pictures (Stanley, 2007). Similarly, by adding the time step as input, the CPPN can produce a time series of patterns, depicting a dynamic structure such as an activation wave, as shown at the bottom of Figure 3.

In the DST model, an ANN's genome encodes various parameters for each of its layers, such as the number of initial neurons. In addition, it can include a set of CPPN genes for any layer such that the decoded CPPN can be used to generate activation waves during development.

Neural layers are modeled as 2d surfaces, where each neuron (n) has a center coordinate, (x_n, y_n) . During development, to compute the wave-induced activity of n at time t , simply input x_n , y_n and t to the layer's CPPN and interpret the output value as a local activation.

When adjacent layers in an ANN include CPPNs, each can be run to produce activation patterns. As shown in Figure 4, when neurons j and k in adjacent layers (J and K) have correlated wave-induced activation, Hebbian-based synaptic tuning on the j - k connection provides an early bias of the network. When the activation waves fortuitously reflect some aspect of the sensory world to which the organism will eventually be exposed, this preliminary synaptic tuning should provide a useful *head start* for the ANN and agent.

Hence, by including CPPN parameters with the other layer-specific genes in an evolving ANN, any pair of interconnected layers with CPPN-based developmental stimulation can achieve an evolving prenatal bias of its weights.

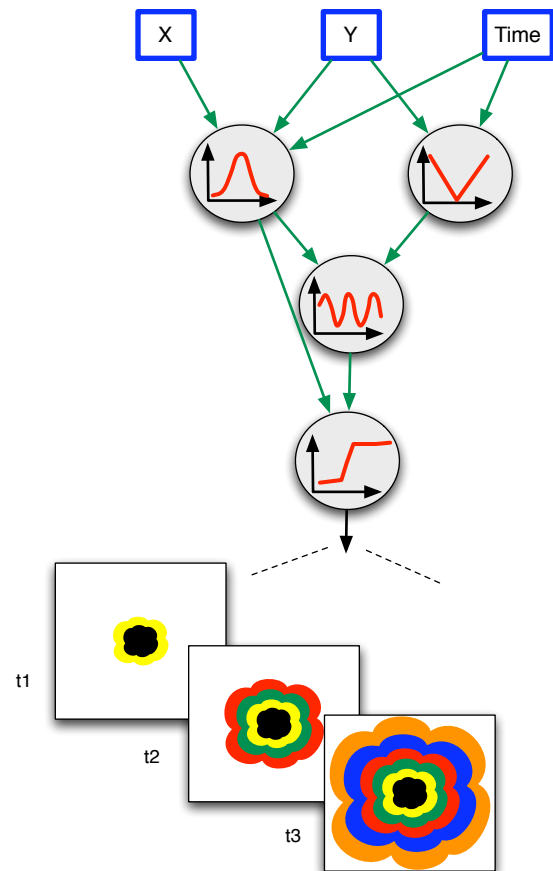


Figure 3: A CPPN, when provided with Cartesian coordinates and time as inputs, produces abstract temporal activation patterns.

Evolving CPPNs

CPPNs, as defined in (Stanley, 2007) are evolved via the NEAT system (Stanley and Miikkulainen, 2002), which has the advantage of supporting gradual complexification but which is a rather direct encoding, with one gene required for each node and weight. This work employs a more generative CPPN encoding to reduce the need for individual weight genes and achieve a bit more biological plausibility.

These CPPNs evolve via a simple bit-vector chromosome consisting of multiple segments, one for each input, internal and output node in the network. Each segment consists of 5 genes that encode the: a) activation function, b) afferent connection tag, c) efferent connection tag, d) afferent weight tag, and e) efferent weight tag.

The first is simply an index into a list of possible activation functions (identity, sine, absolute value, gaussian and sigmoid), while the afferent tags for node N help determine a) which nodes can send input to N , and b) the weights on those incoming arcs. Similarly, the efferent tags influence a)

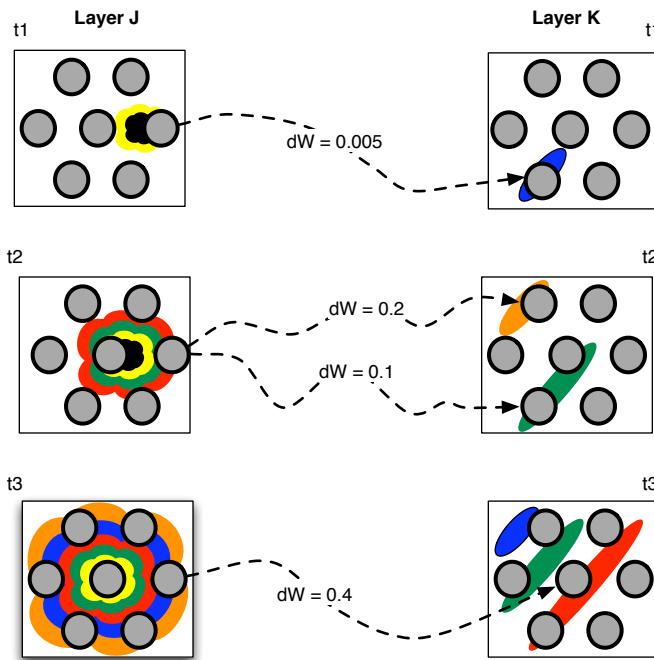


Figure 4: CPPN-generated activation patterns stimulate adjacent neural layers, leading to correlation-based weight changes (dW) to synapses between co-active neurons.

the nodes to which N can send its output, and b) the weights on those outgoing arcs. The two afferent (efferent) tags constitute the afferent (efferent) *mask* of each node.

More specifically, if the afferent connection tag of node N matches the efferent connection tag of node M (above a user-defined match threshold, e.g. 0.75), then M will send an excitatory connection to N . Conversely, if the match is very poor, and thus below a similar threshold, e.g. 0.25, then M will send an inhibitory connection to N . For medium-strength tag matches, no connection between M and N is created. Then, the strength of an excitatory or inhibitory arc is positively correlated with the matching degree of M 's efferent-weight- and N 's afferent-weight tags.

Our CPPNs have a pre-defined number of input and output nodes, but the efferent and afferent masks (along with the activation functions) of output nodes can evolve, as can the efferent masks of each input node. Since recurrent links are permitted in CPPNs, both types of masks are relevant for output nodes, while input nodes are strictly entry ports for external data. During CPPN configuration, once all connections are determined, nodes that either form no connections or, more generally, do not lie along at least one pathway from some input to some output node, are pruned.

Combining DST with Backpropagation

To investigate Heterochronous Neural Baldwinism using DST, we employ a 3-layered (input, hidden, output) ANN, with 9 linear input, 9 sigmoidal output, and a variable number of sigmoidal hidden nodes (coded in the genome). Standard backpropagation (BP) accounts for all of the learning, while CPPN-generated waves handle developmental tuning of connections between the hidden and output neurons. The complete DST-BP process is summarized in Figure 5.

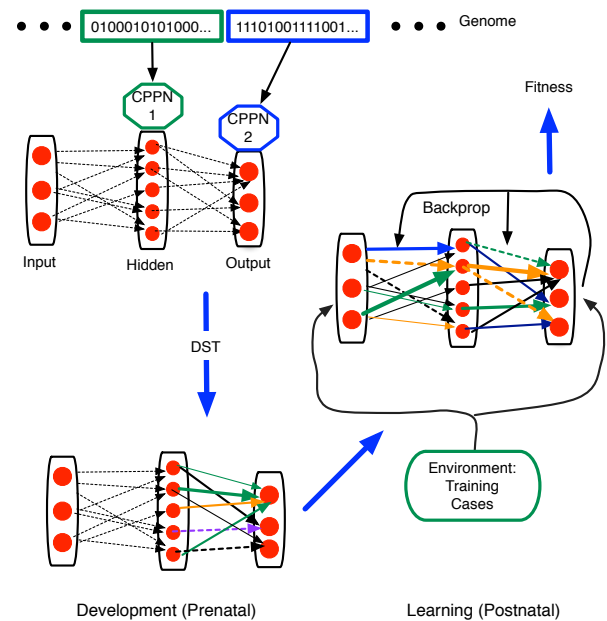


Figure 5: Key stages of the DST-BP process: (Upper left) Translation of chromosomal segments to CPPNs for 2 layers; (Lower left) DST provides an initial bias to hidden-to-output connections; (Right) BP learning further tunes all synapses to capture the training set, with fitness based on training error, test error, and BP learning effort.

The Genetic Algorithm

In the spirit of Hinton and Nowlan's original work, we use relatively small populations evolved over relatively few generations to solve simple problems, primarily as a proof of concept. The key GA parameters include a population size of 20, full-generational replacement with a rank-based selection mechanism and elitism of two individuals, a crossover rate of 0.8 and a mutation rate of 0.05 per bit.

The GA chromosome for DST-BP involves 9 basic developmental and learning parameters, while two CPPNs (for the hidden and output layers) require 14, 5-part genes apiece (encoding activation functions and masks). The 3 developmental parameters are: 1) initial hidden-layer size ($H_{init} \in [1, 10]$), 2) developmental tuning rate ($D_r \in [0, 1]$), and 3)

activation wave steps ($D_s \in [0, 5]$). The 6 learning parameters are 3 each for the input-hidden layer connections, and the hidden-output layer links. The former 3 are: a) learning rate, $L_{r1} \in [0, 1]$, b) learning epochs, $L_{e1} \in [1, 10]$, and c) momentum, $M_{e1} \in [0, 0.2]$, while the latter 3 are: d) $L_{r2} \in [0, 1]$, e) $L_{e2} \in [1, 10]$, and f) $M_{e2} \in [0, 0.2]$.

Fitness Testing

Data sets for backpropagation learning are generated by one-dimensional cellular automata (CA), run for 10 timesteps, with each consecutive pair of 9-cell states constituting a training case, i.e., $s_t \rightarrow s_{t+1}$, where s_t , the CA state at time t , is loaded onto the input neurons, with the target output being s_{t+1} . The key point is that if a bit is on (or off) in s_t , this influences the chances of it and other neighboring bits being on or off in s_{t+1} , thus adding some structure to the data.

Fitness stems from both training and testing error, with the former consisting of the average error per output neuron, per training case, per epoch; while the latter is per neuron per case for a single epoch, without learning. Thus, the abilities to a) *quickly* reduce error during training, and b) *eventually* reduce that error, are independently assessed.

In all of the runs reported below, each individual undergoes 5 independent rounds of fitness testing, wherein a different set of random initial weights (in the range $[-0.2, 0.2]$) are assigned. The error terms E_{train} and E_{test} denote averages over these 5 rounds, and all runs employ a *tuning tax*, $\Theta = 0.04$. The fitness function of equation 1 accounts for both error terms along with the learning effort:

$$f = e^{-(E_{train} + E_{test} + \Theta L_e L_r)} \quad (1)$$

where $L_e = \min(L_{e1}, L_{e2})$ (since only this minimum of the two values of epochs is actually performed), and $L_r = \frac{L_{r1} + L_{r2}}{2}$.

The Developmental Contribution

In each of the runs below, the contribution of developmental synaptic tuning to error reduction is estimated by a simple test, performed only on the best-of-generation individuals. First, the weights of the ANN are randomly initialized, before sending the entire training set through the network, but without learning. The average error, per output node, per training case, is then compared to the average error in a second test, wherein the same ANN, with the same initial weights, also undergoes the developmental tuning encoded in the genome. Differences in these two error terms gives a rough indication of the contribution of DST to error reduction, and thus to fitness. It is important to note that the fitness function does not explicitly reward this contribution. Its effect is only indirect, via the reduction in learning effort afforded by DST.

In the runs below, a typical training error prior to any tuning (developmental or learning) is 0.35 to 0.45, while the

improvement typically varies from 0 to 0.05 (e.g. $0.45 - 0.40 = 0.05$).

Results

For each of the runs presented below, 5 properties of the best-of-generation individual are plotted: fitness, learning effort, training error, test error, and developmental contribution to error reduction. To easily display each value on the same linear plot, the two error values and the developmental contribution are multiplied by 10.

Figure 6 illustrates a sample 300-generation run using DST-BP and a CA-generated data set. HNB phase I involves a gradual increase in learning effort over approximately 80 generations, during which fitness rises and error falls. Phase II begins thereafter, with learning effort dropping in several discrete steps over the final 220 generations, although the lowest learning-effort value appears evolutionarily unstable. In this plot, the rise of fitness during phase II is clearly evident. The developmental wave functions for the best-fit individual of the 300th generation are shown in Figure 7.

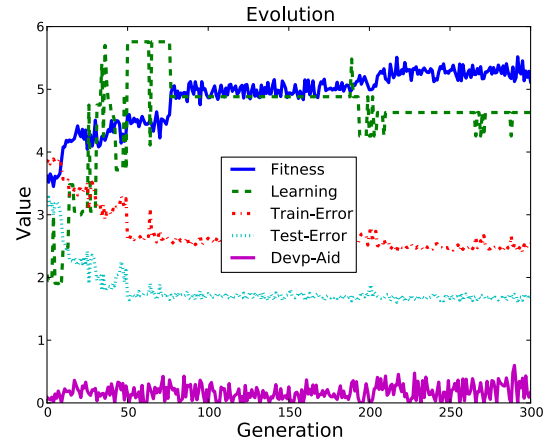


Figure 6: Time series of fitness, learning effort, two error terms, and developmental enhancement for each best-of-generation individual in a 300-generation DST-BP run.

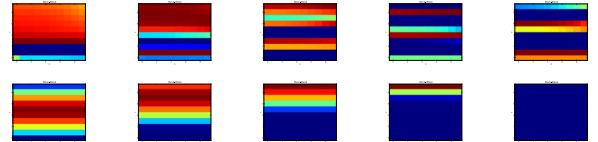


Figure 7: The 5-step, CPPN-generated, developmental activation waves for hidden (top row) and output (bottom row) layers for an evolved, 3-layered feed-forward network using backpropagation learning. Red (blue) indicates maximal (minimal) stimulation.

To test the generality of HNB using DST-BP, a series of 20 independent runs are performed, each using a different CA-generated dataset. The best-of-generation averages, shown in Figure 8, display the general Baldwin Effect in that learning initially boosts fitness, but then plasticity decreases while the error terms remain lower (than the early-generation values), and fitness gradually increases.

Heterochronous Neural Baldwinism is evidenced by the bottom line in the figure, which shows a gradual rise in the contribution of development to error reduction. This rise is barely perceptible in the figure, but a comparison of the first 50 averages (over 20 runs) to the last 50 shows a statistically significant difference ($p = .0005$) in a single-tailed Student-t test: the averages are 0.020 and 0.027 for the first and last quarter, respectively. This indicates that the developmental contribution to output error reduction, though small, does allow learning effort to decrease.



Figure 8: Averages (over 20 independent runs using different CA-generated datasets) of fitness, learning effort, two error terms, and developmental enhancement (devp) for best-of-generation individuals in a population of 20 DST-BP networks.

In another set of 20 runs, the datasets consist of randomly-generated sparse patterns, with exactly 3 ones and 6 zeros in each. In contrast to the CA-generated datasets, these have no spatial relationship between the on (1) bits of the input and output/target patterns, thus making it harder for evolution to find helpful developmental schemes. However, HNB occurs in these runs (not shown) as well, with gradually declining learning effort and gradually increasing (and statistically significant, $p = .0005$) developmental contribution and fitness.

Further evidence for the contribution of development appears in Figure 9, which displays 20-run averages for scenarios using CA-generated datasets, but no DST. Notice that the learning effort rises and remains high throughout the

200 generations. Without developmental assistance, learning must remain elevated to keep the error levels in check. This continuously-high learning cost keeps fitness levels in Figure 9 well below those of Figure 8.

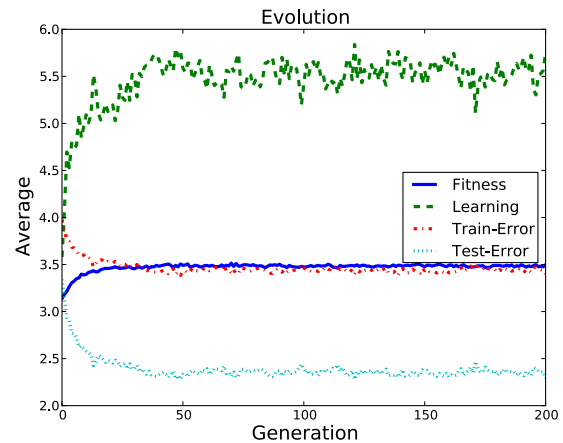


Figure 9: Averages (over 20 independent runs using different CA-generated datasets) of fitness, learning effort, and two error terms, for best-of-generation individuals in a population of 20 DST-BP networks, where DST is silenced.

Discussion

The DST-BP model gives preliminary evidence of the Baldwin effect in neural networks that undergo a developmental process. The validity of these results hinges on a quantitative (rather than qualitative) interpretation of the key differences between development and learning in neural systems. Namely, cross-generational changes in the pre- and postnatal **rates** of neurogenesis, synaptogenesis and synaptic tuning can transfer adaptive effort from learning to development, with the latter more closely governed by the genome and less by environmental factors. Thus, B.E. phase 2 transfers a portion of brain formation backwards, from learning to development, and thus to a stage where it is more strongly affected by the genome - and can thus lay claim to being *more innate* than characteristics acquired later in life, when environmental influences typically play a more decisive role.

As in (Downing, 2004), the effects of learning are not reverse-encoded into the genome, but strong learning *buys evolutionary time* until proper developmental scaffolding reduces the overall postnatal adaptive costs, thereby raising fitness to peak levels. In this work, scaffolding involves CPPN-generated activation waves, and the ensuing prenatal synaptic tuning, which provides the postnatal phase with a synaptic matrix that is already partially biased toward the environment (i.e. training set).

The DST-BP model embodies a complex mapping from genes to synaptic weights (via CPPNs and DST), which

might seem to preclude B.E. phase II. However, the genome possesses enough flexibility to evolve developmental scaffolding capable of reducing postnatal adaptive demands. Thus, although the *results* of postnatal synaptic tuning do not become innate, their *attainment* becomes easier due to evolved innate processes.

The DST model says little about the potential applicability of CPPNs as activity-wave generators for evolving ANNs, and it seems impractical to go to such lengths to produce functioning ANNs for complex tasks. But in support of HNB, the CPPN provides an appropriate abstraction (over complex reaction-diffusion interactions) for generating activity waves whose biological counterparts do appear to play an important role in early neural circuit formation. The DST-BP model indicates that these activity waves may also help explain Neural Baldwinism, though a more convincing argument would involve a learning mechanism of greater biological realism than backpropagation. To this end, we have combined DST with Hebbian learning in simple two-layered ANNs. This produces a more dramatic Baldwin Effect than the DST-BP runs, both in terms of a greater learning declines and very sizeable developmental contributions to error reduction (often over 30%). However, the Hebbian model was only able to learn very simple training sets involving very sparse input and output vectors.

Despite these relatively weak results, ALife researchers should profit from this article's primary insight: the assimilatory phase of the Baldwin effect, when viewed as a quantitative rather than qualitative shift in activity from later to earlier stages of life, does reduce the general complexity (and near impossibility) of the transfer of the fruits of neural plasticity to the realm of genetic control.

As elaborated by several influential biologists (West-Eberhard, 2003; Kirschner and Gerhart, 2005), the interactions between evolution, development and learning are intricate and multifaceted. Though often intimidating, these complex relationships may in fact open the gate for interpretations of B.E., such as HNB, that can enhance its general plausibility with respect to the evolution of intelligence.

References

- Ackley, D. H. and Littman, M. L. (1992). Interactions between learning and evolution. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II*, pages 487–509, Reading, Massachusetts. Addison-Wesley.
- Baldwin, J. M. (1896). A new factor in evolution. *The American Naturalist*, 30:441–451.
- Bull, L. (1999). On the Baldwin Effect. *Artificial Life*, 5(3):465–480.
- Downing, K. L. (2004). Development and the Baldwin Effect. *Artificial Life*, 10(1):39–63.
- Downing, K. L. (2010). The Baldwin effect in developing neural networks. In *Proceedings of the 12th Genetic and Evolutionary Computation Conference*, pages 555–562, Portland, Oregon. ACM Press.
- Gruau, F. and Whitley, D. (1993). Adding learning to the cellular development of neural networks. *Evolutionary Computation*, 1(3):213–233.
- Hinton, G. E. and Nowlan, S. J. (1987). How learning can guide evolution. *Complex Systems*, 1:495–502.
- Kirschner, M. W. and Gerhart, J. C. (2005). *The Plausibility of Life: Resolving Darwin's Dilemma*. Yale University Press, New Haven, CN.
- Mayley, G. (1996). Landscapes, learning costs and genetic assimilation. *Evolutionary Computation*, 4(3):213–234.
- McNaughton, B., Battaglia, L., Jensen, O., Moser, E. I., and Moser, M. B. (2006). Path integration and the neural basis of the 'cognitive map'. *Nature Reviews Neuroscience*, 7(8):663–678.
- Paenke, I., Kawecki, T., and Sendhoff, B. (2009). The influence of learning on evolution: a mathematical framework. *Artificial Life*, 15(2):227–245.
- Sanes, D., Reh, T., and Harris, W. (2006). *Development of the Nervous System*. Elsevier Academic Press, Burlington, MA.
- Shors, T. J. (2009). Saving new brain cells. *Scientific American*, 300(3):40–48.
- Stanley, K. (2007). Compositional pattern producing networks: a novel abstraction of development. *Genetic Programming and Evolvable Machines: Special Issue on Developmental Systems*, 8(2):131–162.
- Stanley, K. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.
- Stellwagen, D. and Shatz, C. (2002). An instructive role for retinal waves in the development of retinogeniculate connectivity. *Neuron*, 33(1):357–367.
- Suzuki, R. and Arita, T. (2007). The dynamic changes of roles of learning through the Baldwin Effect. *Artificial Life*, 13(1):31–43.
- Turney, P., Whitley, L. D., and Anderson, R. W. (1997). Introduction to the special issue: Evolution, learning, and instinct: 100 years of the Baldwin Effect. *Evolutionary Computation*, 4(3):iv–viii.
- West-Eberhard, M. J. (2003). *Developmental Plasticity and Evolution*. Oxford University Press, New York, NY.

A quantitative measure of non-neutral evolutionary activity for systems that exhibit intrinsic fitness

Alastair Droop¹ and Simon Hickinbotham²

¹Leeds Institute of Molecular Medicine, St James's University Hospital, Leeds LS9 7TF, UK

²NSC and ACA groups, Dept. of Computer Science, University of York, Heslington, York YO10 5GH, UK
sjh436@gmail.com

Abstract

Open-ended evolutionary systems offer us the tantalising prospect of creating artificial life from simple precursors. One of the issues in designing open ended systems is that there exist few metrics for measuring their evolutionary activity. Current measures of evolutionary activity are only applicable to systems in which the fitness of a single component is defined by an explicit fitness function. However, this is not guaranteed in systems where a significant part of the fitness is intrinsic (for example caused by interactions between components). In this paper, we evaluate a new approach to the problem of measuring evolutionary activity that is applicable to systems exhibiting both explicit and intrinsic fitness pressures. To evaluate this measure, we ran 22,000 grid-based simulations of two automata chemistries, Tierra and Stringmol. Both of these systems have strong intrinsic fitness pressures. We examine the effect of varying the mutation rate in both systems, and demonstrate that the new measure identifies an optimal mutation rate.

Introduction

We are interested in the design of automata chemistries (AChems) that are capable of open-ended evolution. One of the issues in this arena is that it is very difficult to determine whether a design change has delivered an increase in open-endedness. This issue is well-known and has been studied since the beginnings of the ALife paradigm.

Evolution acts upon definable entities (*components*) in a system. A component is defined as any part of a genome, but is commonly used in ALife to refer to the entire genome of an individual. In ALife, a species can be a unique sequence of codes, and all entities in a system with the same sequence are assigned a single component label. In real biology, a *species* is to (a great extent) analogous to a component. Evolutionary activity measures based upon population dynamics can therefore be applied to both biological and ALife systems as long as a consistent definition of a component is used.

Evolutionary systems generate novel components by mutation. This process does not guarantee to increase fitness. Furthermore, many systems do not have simple genotype–phenotype mappings, and many different components can

have the same overall fitness. Different components with similar fitnesses exhibit *neutral drift* in their population sizes, as there is not sufficient difference in their fitness for one component to gain a significant selective advantage over another. Neutral drift is observable in both biological and ALife systems, and appears as a random walk. A measure of the propensity of a system to give rise to mutations that increase the overall system fitness (the *Evolutionary Activity* or *EA* of the system) is required to guide the design of ALife systems.

There are established ways of characterising the EA of a system by studying the population dynamics of new variants as they evolve through time (Bedau et al., 1997; Ray and Xu, 2001; Bullock and Bedau, 2006; Channon, 2006). There are two problems with these measures: the first is that although they allow qualitative assessment of EA they do not deliver a numerical measure that can be used to compare designs; the second is that the method of accounting for neutral mutation requires that neutral mutations can be accessed or generated by the system whenever needed. This requirement implies that the fitness of the individual can be known completely, and the effects of fitness on selection can be negated. Yet ALife simulations, particularly “automata chemistries” (Dittrich et al., 2001) of any sophistication, tend to exhibit *intrinsic* fitness that cannot be factored out when neutral variants are required.

EA measures seek to identify components that have some new behaviour that confer a fitness advantage in the prevailing conditions. Such measures must discriminate between innovation (the production of novel components with a selective advantage) and neutral drift.

Early attempts to quantify neutral drift used secondary (*shadow*) simulations identical to the system under study (the *foreground* system) in which selective pressure has been removed, and thus all selection is neutral. However, over time the shadow and foreground systems diverge, making comparisons between them uninformative. For example, Ray and Xu (2001) tried to run a shadow simulation on Tierra (Ray, 1991), but were unable to identify a measure that was capable of satisfactorily capturing the EA. The ap-

proach was extended by Channon (2006), who argued for continuous re-setting of the shadow throughout a run. These re-setting activities are expensive, and so are commonly not calculated for each time point. Furthermore, the shadow simulation approach is not applicable when studying systems with intrinsic fitness.

Systems with intrinsic fitness are not uncommon. In real biology, there is no explicit fitness function, thus all fitness is intrinsic. Similarly, the Stringmol AChem (Hickinbotham et al., 2010a,b) has no fitness function: the fitness of a component is defined solely by the dynamics of its interaction with other components. Tierra uses some small fitness measures to contribute to the ordering the “reaper” and “slicer” queues, but these pressures are by no means the only ones affecting survival. For example, in both of these AChem systems, resistance to parasitic attack is an important part of the phenotype of the individual.

Historically, EA measures were studied over small numbers of runs. These early measures allowed qualitative views of a trial to be created, and allowed innovative components in a trial to be identified quickly. The focus on individual runs was partially due to the computational overheads that are required to run these simulations, especially in work from over a decade ago. With the advent of cheap and fast grid computing, we are in a better position to research EA at the system level. It is straightforward to observe that most open-ended AChem systems follow very different trajectories between multiple runs due to their stochastic nature. Measures of a EA at the system level must therefore be based upon multiple runs, precluding the use of qualitative assessments. We require a single, numerical evaluation of a complete run, that forms a statistic for measuring the evolutionary power of the system design. In this respect, our work is different from previous contributions. We believe that measures such as these will form more powerful (statistical) indicators for AChem and ALife designers.

If we are designing systems that manage their own evolution, it will be important to have a quantitative measure of EA to allow comparisons to be made between configurations of a system, and in due course to improve the design of the system. Below we present an alternative means to accommodate neutral drift in measures of evolutionary activity. The new measure is based on the idea that systems of species which differ only in neutral regions of their genome will exhibit random walks in their populations.

Evolutionary Activity measures

The practicality of established methods for measuring EA are one their key strengths. It is usually straightforward to record the time, component type and component count of a simulation, and this is all that is required to develop an analysis of EA. The authors of this contribution have developed software in the R programming environment that reads in this data from a simple comma delimited list, and allows the

figures and statistics presented here to be produced easily (Droop and Hickinbotham, 2011c). Two established measures of EA are available, along with the new method we develop here.

For EA measures to be comparable across different systems, it is desirable that they are based on observable characteristics that are present across a wide range of ALife and Biological systems. Accordingly, EA measures (including our proposed approach) tend to be based on counts of individual belonging to a particular component (species) $i \in I$ across discrete time samples $t = 0 \dots T$, where T is the total time that the system is run for and I is the set of different components observed in the system.

Framework

An excellent summary of the EA measure is presented in Channon (2006), whose notation we follow here. The premise is that a component’s activity should accumulate by some measure Δ , called the *activity increment* for every time sample in which it is observed. In order to generate summary statistics for a system, values of Δ are obtained for every component. Summaries can be obtained over time steps, over components, or over a combination of both. There are several ways to calculate the value of Δ . Here we describe methods based on the presence of a component, Δ^P , and methods based on population counts Δ^C .

The earliest and simplest formulation Δ^P is a Boolean function over the presence of a component i at time t :

$$\Delta_{i,t}^P = \begin{cases} 1 & \text{if component } i \text{ is present at } t \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The idea is that the utility of the component is reflected in its longevity (the number of time steps for which it is present in the simulation).

It is clear that Δ^P takes no account of the numbers of individuals at each time step, so an alternative, Δ^C uses counts ($c_{i,t}$) of individuals for each time step:

$$\Delta_{i,t}^C = \begin{cases} c_{i,t} & \text{if component } i \text{ is present at } t \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

For any increment Δ , the activity a is calculated using:

$$a_{i,t} = \begin{cases} \sum_{\tau=0}^t \Delta_{i,\tau} & \text{if component } i \text{ exists at } t \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Note the phrase “exists” has a special meaning here — a component exists in all time steps between the time it is first observed, and the time that it is last observed. Since a gives activity per component, the total cumulative evolutionary activity A_t at each time step is:

$$A_t = \sum_{i \in I} a_{i,t} \quad (4)$$

Applying a shadow model

As stated above, measurement of EA using shadow simulations is not appropriate in systems with intrinsic fitness. If the shadow approach is utilised, the shadow run allows us to estimate the neutral activity a^N , and this allows us to estimate the adapted activity a^A by:

$$a^A = a^* - a^N \quad (5)$$

Where a^* can be either a^P or a^C .

This approach neglects the contribution to fitness of an individual that is afforded by other components in the system. This effect is not negligible in automata chemistries, where the process of interaction between individuals is an essential part of the system. Furthermore, steady-state periods increase the values of a for both measures. Whilst this is useful for interpreting individual plots (since it emphasises the more dominant components), it is not itself a measure of EA. With these observations in mind, we develop a quantitative measure of *non-neutral* EA below that does not require the use of a shadow model.

Non-Neutral Evolutionary Activity

Our measure is based on the observation that if mutations in a population were neutral, then the size of the population of each component would follow a random walk from one time point to the next. On average, the population sizes of a neutral variant of a component should be approximately equal between neighbouring time points — there will of course be some small variation around this average. Further, we can assume that any marked increase above the predicted population size is a good indicator that a component has some fitness advantage that allows it to increase in number, commonly at the expense of other components.

Using the notation given above, we develop a new activity increment Δ^N . Firstly, we calculate the total population C at time t :

$$C_t = \sum_{i \in I} c_{i,t} \quad (6)$$

and use this to calculate the species proportions p_i at t :

$$p_{i,t} = c_{i,t}/C_t \quad (7)$$

The expected proportion e_i of a component is simply the proportion observed in the previous time step:

$$e_{i,t} = \begin{cases} p_{i,t-1} & \text{if } t > 0 \\ 0 & \text{if } t = 0 \end{cases} \quad (8)$$

The new activity measure is the square of positive values of $p - e$ scaled by the total population at t :

$$\Delta_{i,t}^N = \begin{cases} C_t(p_{i,t} - e_{i,t})^2 & \text{if } p_{i,t} > e_{i,t} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

In this metric, EA is encapsulated as the difference between $p_{i,t}$ and $e_{i,t}$. By squaring the difference between $p_{i,t}$ and $e_{i,t}$

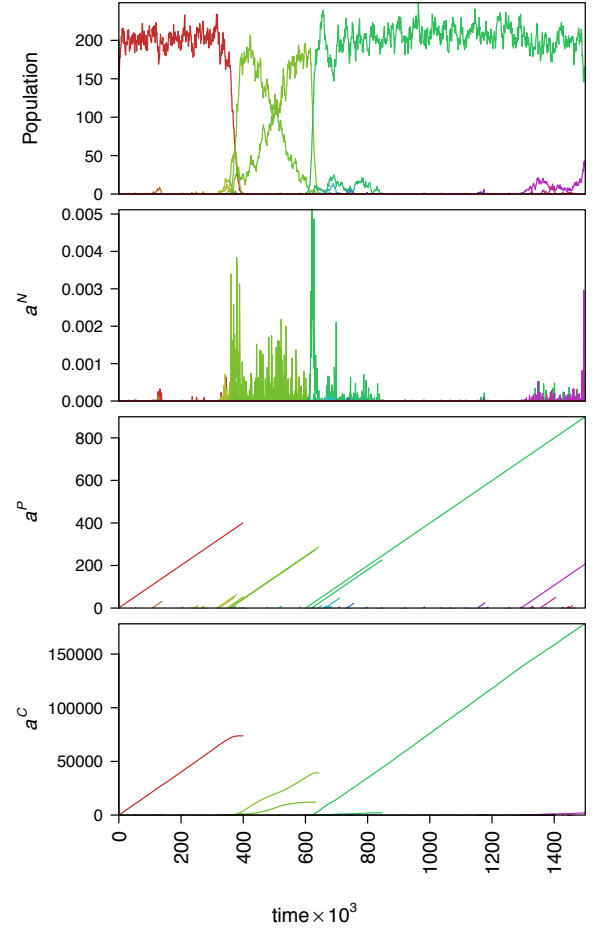


Figure 1: Component population dynamics (top), with a^N (second row), a^P (third row) and a^C (bottom) measures.

(reminiscent of simple statistics of variation and concepts of inertia), we emphasise these differences.

Figure 1 shows the change in value of Δ^P , Δ^C , and Δ^N for a period of evolutionary activity in a run of the String-mol AChem. The population dynamics of each component are shown in the top plot. There are three phases in the plot. The first phase has a single dominant species (shown in red). This period ends when two new components arise, one of which increases rapidly, and one of which increases slowly. These components are themselves replaced by a new dominant species that remains dominant for the rest of the time shown in the plot. Below this plot, figure 1 displays (from top to bottom) changes in a_i for all components based on Δ^N , Δ^P , and Δ^C respectively. It is clear that both Δ^P , and Δ^C (the two lower plots) emphasise the dominance of particular component types, but Δ^N (second from top) specifically highlights the EA caused by the introduction of new components. We believe that this is a more useful measure of evolving systems, since periods of (relative) stasis do not score highly with Δ^N .

Experiments

In order to demonstrate the EA measures at the system level, we selected two automata chemistries (Tierra and Stringmol) and calculated the EA for a range of different mutation rates. We chose Tierra because it is arguably the most famous automata chemistry in ALife, and has been referenced in Bedau et al's original work. Stringmol represents a very different set of design choices, and was selected to test the applicability of our EA measure to multiple AChems. The mutation rate is an obvious variable to experiment with in most ALife scenarios for multiple reasons: not only is it key in any evolving system, but it is likely to exhibit an optimum rate (low values lead to stasis, whilst high values cause error catastrophe (Eigen, 2002)).

Due to the stochastic nature of mutations (especially at low mutation rates), multiple runs of each system are required to generate reliable statistics. In this work, each system was run 100 times for each parameter set utilising the Volvox grid at the University of York. Each simulation was allowed to run for a maximum of 1×10^9 time steps. The authors have previously carried out large scale simulations with Stringmol (Droop and Hickinbotham, 2011a), but these experiments did not use EA measures.

Tierra Configuration

We used Tierra 6.02 in our experiments, but applied the patch developed by Rav (2011). We also edited the source code to deliver a file in a format that is compatible with our R analysis libraries. These are available on the authors' website (Droop and Hickinbotham, 2011c).

For a detailed overview of Tierra, see Ray (1991). A Tierra simulation consists of a set of individual "programs" existing in a "soup" of memory. The individual in Tierra is a string of opcodes and a set of registers. The opcodes specify a sequence of computational operations that shifts values between the opcodes and the registers in a manner similar to conventional computers. The individuals compete for processor time, which is allocated via a control structure called a *slicer*. Another controller, the *reaper* determines whether an individual should be deleted or not at each time step. The system is seeded with a set of individuals containing codes that instruct the system to perform operations on locations in memory in order to create a copy of the individual. The locations of the copies are determined by shifting pointer addresses around according to some match function. Individuals are not easily able to defend themselves against being overwritten, but since pointers are moved by a match function, it is advantageous for the individual to have nothing for the match function to match with. In addition, the shorter the individuals instruction set is, the more rapidly it can be copied.

We made three changes to the canonical Tierra system and the "gb0" simulation that comes with the software. The first of these was to set the mutation rate to a specific value

for each simulation, rather than making it a variable that changes during run-time as a function of the average component length. This was done to ensure that there was no danger that drifting mutation rates would mask the changes that the activity measure was designed to detect. The second change was to set all mutation rates to zero except for the "RateMovMut" parameter, which was varied by seven orders of magnitude. The third change allowed us to increase the mutation rate up to a point where no symbol could be copied successfully, whereas the original code only allowed a maximum rate of one mutation per component. See table 1 for the mutation values we used. Additional changes were made to the code to gather statistics, but none of these interfere with the function of the program.

Stringmol Configuration

Stringmol is a modern automata chemistry designed to be much simpler than its forebears by placing less emphasis on registers and memory addressing, and more emphasis on the process of binding as a precursor to a reaction between components. An individual in Stringmol consists of a string of opcodes and four program pointers. There are no queues for processor time or death — both of these are selected stochastically. Components survive and multiply by copying themselves more quickly (on average) than they are destroyed. Individual components only run their programs when they bind to other individuals, meaning that an individual has no opportunity to interfere with its neighbours, *unless* it can bind to it. This was designed to emulate the specific binding properties of enzymes and substrates, and makes the system much less noisy than Tierra. In these experiments, we used the configuration of Stringmol as described in Hickinbotham et al. (2010a). Although we set a limit of 1 billion time steps on the simulations, this was never reached because simulations in this configuration always terminate due to parasitism.

Mutation Rates

The value of the mutation rate requires special interpretation in both Tierra and Stringmol. In Tierra, the "RateMovMut" parameter specifies the number of mutations that will occur when a component of a particular length is copied. For example, a rate of 32 would mean that on average, 1 in 32 copies of a component would contain a mutation. But if the lengths of components changes, then this rate would change. To allow us to compare Tierra with Stringmol, we chose to specify mutation rate in terms of the fraction of mutations per copy operation. A value of 0.1 indicates that one mutation will occur every 10 operations. We can easily map this value to "RateMovMut" in Tierra, using twice the component length of 80.¹ The mutation rates used in this work are

¹Note that we found a scalar of 2 for the mutation rate in the source code for Tierra 6.02, but could find no reference to its significance.

m	m^{-1}	R	standard value
1.000	1	-	
5.000×10^{-01}	2	-	
2.500×10^{-01}	4	-	
2.000×10^{-01}	5	-	
1.000×10^{-01}	10	-	
5.000×10^{-02}	20	-	
2.500×10^{-02}	40	-	
1.250×10^{-02}	80	-	
6.250×10^{-03}	160	1	
3.125×10^{-03}	320	2	
1.562×10^{-03}	640	4	
7.812×10^{-04}	1280	8	
3.906×10^{-04}	2560	16	Tierra
1.953×10^{-04}	5120	32	
9.766×10^{-05}	10240	64	
4.883×10^{-05}	20480	128	
2.441×10^{-05}	40960	256	
1.221×10^{-05}	81920	512	Stringmol
6.104×10^{-06}	163840	1024	
3.052×10^{-06}	327680	2048	
1.526×10^{-06}	655360	4096	
7.629×10^{-07}	1310720	8192	
3.815×10^{-07}	2621440	16384	

Table 1: Mutation rates used in the Tierra and Stringmol trials. m is the number of mutations per opcode copy. R is the value for the Tierra variable “RateMovMut” that induces mutations at the same rate. The standard values of mutation in Tierra and Stringmol are shown in the right-hand column.

given in table 1.

Results

Tierra

The number of components and values of A^N , A^P and A^D for 100 runs of Tierra at each of 22 different mutation rates are shown in figure 2. A run with the standard (“si0”) settings of Tierra was also done to allow comparison — this is shown as the left hand column of each plot in the figure. The effect of the mutation rate on the number of component types in the system increases up to $m^{-1} = 80$, and then declines to a steady rate when $m^{-1} > 5$. This fits with the idea that up to a point, mutation increases diversity; but beyond that the system loses its ability to self-maintain so that nothing *but* diversity remains, with little functional structure available to perform any useful action. The peak in diversity is important, because it should set an upper bound on the value of m within which EA should be maximal: if the EA measure is maximal above this, then it is just measuring noise. We would expect that a good EA measure would peak somewhere between the point where it is clear that the mutation rate is so low that long periods of stasis are evident and the point where the diversity is at its highest.

The A^N measure on the Tierra trial exhibits these properties, whereas A^P and A^C show positive and negative cor-

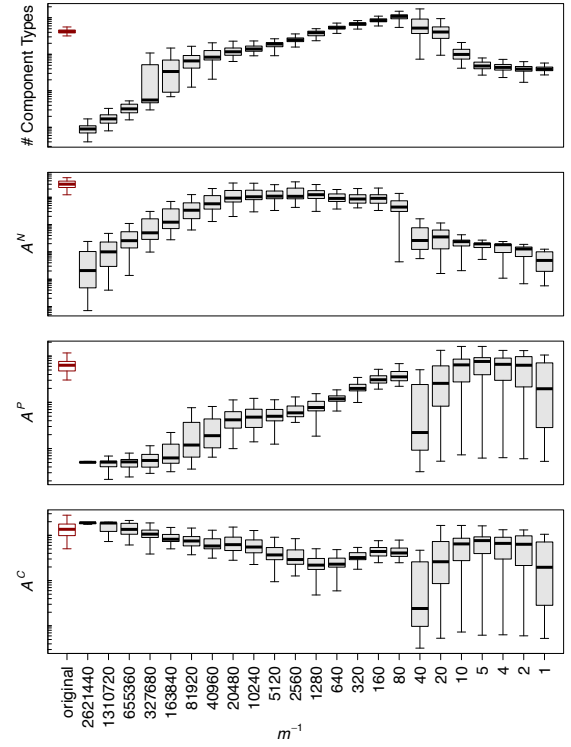


Figure 2: Effect of change in mutation rate in Tierra between the range $m = 2621440^{-1}$ and $m = 1$ mutations per base copied. From top to bottom: Number of species produced in a run; A^N , A^P and A^C . EA measures for the original mutation configuration of Tierra are shown on the left.

relations with m^{-1} respectively. Figure 2 indicates that the A^N value is maximal at a mutation rate corresponding with 1280 copies per error. This value is in line with our expectations — neither low enough to cause long periods of stasis nor high enough to cause errors to damage the system’s ability to self-maintain. This is good evidence that the new EA measure has some utility in designing ALife systems.

The EA values for the original Tierra settings (left hand boxplots in figure 2) are interesting since they are approximately level with the highest-scoring value for mutation in the single mechanism we used in our experiments. Thus the default parameter settings, with many different mutation mechanisms offers little improvement over a single mutation mechanism. This implies that Tierra has many mechanisms that are superfluous to the core evolutionary activity, and shows that the new measure gives us an opportunity to develop systems including Tierra in a more principled manner rather than via *ad hoc* emulation of biological processes.

Figure 3 Shows the median-scoring run out of the set for the mutation rate with the optimum A^N score ($m^{-1} = 8$, centre), and for mutation values five increments lower (left) and higher (right) than this optimum. This also illustrates

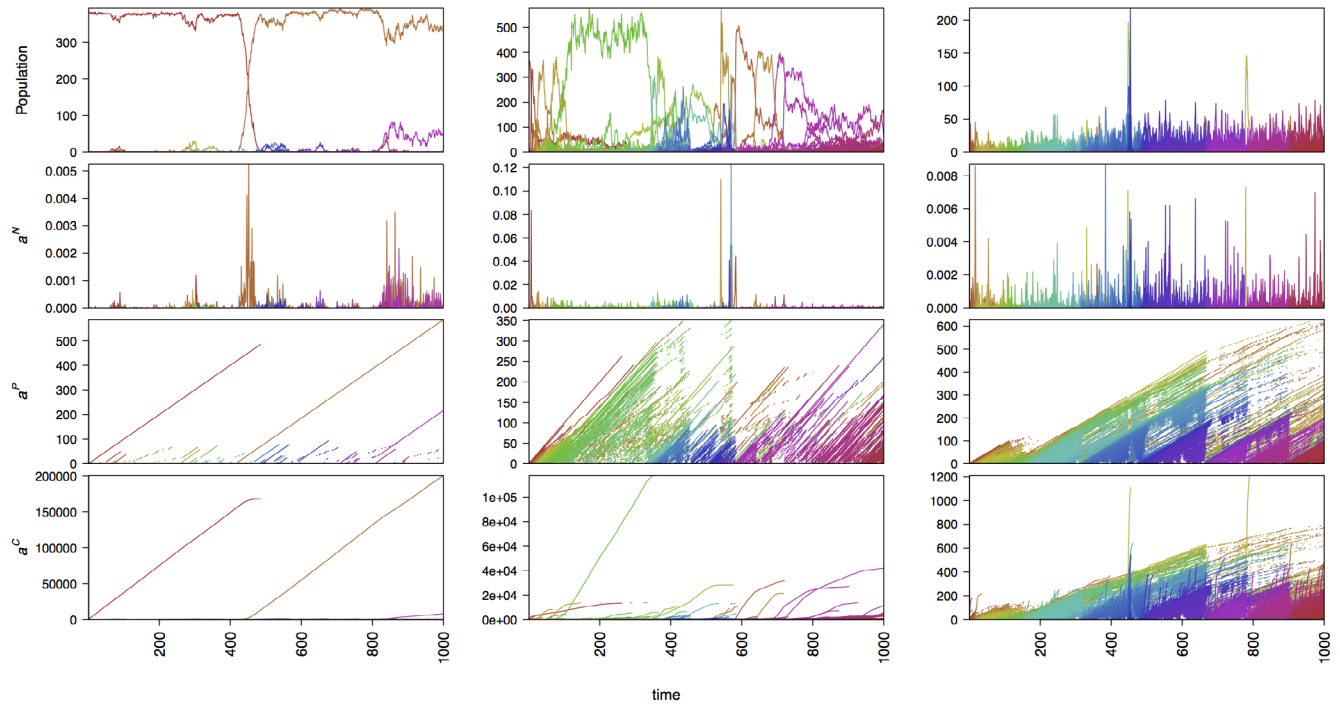


Figure 3: Illustrations of median-scoring runs from the set of 100 trials of Tierra at values of m for which median A^N was maximal (centre column), with the median-scoring run from 5 mutation increments lower (left) and higher (right) than the maximum.

that the measure is effective, since we can see that the lower mutation rate does demonstrate long periods of stasis, whereas the higher mutation rate shows perpetual turnover of component types in smaller populations during the trial. Although A^P and A^C also indicate this graphically, the summation of the values in the plots does not capture the values that allow the optimum mutation rate to be evaluated without costly visual examination of the plots.

Stringmol

We find similar results with Stringmol as we did with Tierra, as shown in figure 4. In Stringmol, parasitism can halt the system completely, so the lifetime of trials is much more variable than for Tierra. Due to the fact that Stringmol does not have a fixed runtime, we divided by runtime to normalise the scores for all EA measures. If we do not normalise the A^N measure by the lifetime of the trial, we find emphasis on more stable runs (figure 4), whereas if we divide by lifetime, we find emphasis on runs with a short lifetime and rich dynamics. These are shown in figure 5. Figure 6 shows typical plots for Stringmol as described for Tierra in figure 3, above. We show four columns, for values of m^{-1} ranging from 2,621,440 to 160. It is clear that the A^N measure is responding to mutation rates which promote evolutionary activity, but we suggest that the effect of parasitism is so great that it skews the new EA measure towards a higher muta-

tion rate than the optimal. This compares favourably with A^P and A^C , which score very low mutation rates ($m^{-1} = 2,621,440$) highly. The measures respond to periods of extreme stasis in Stringmol, rather than the evolutionary activity that we seek to identify.

Broadly speaking then, we find the picture is the same as for Tierra: the new EA measure is capable of detecting runs that match a rate that appears intuitively correct — sufficient diversity, with a balance between stasis and catastrophe. We believe that these data support our assertion of the effectiveness of our Non-Neutral approach to measuring EA.

Comparing Stringmol with Tierra

Our analysis has also highlighted some issues with the AChems that we studied. Tierra runs rarely terminate, even when there is no significant reproduction occurring. Although Tierra has a facility for terminating runs when copy operations are not observed, at high mutation rates it is more common that some copy operations are happening, but they are failing to produce any new individuals since their surrounding apparatus is not sufficiently well realised. We tend to see that the reaper queue is never utilised because there is no pressure on space (i.e. memory) in the environment in which the organisms exist. Tierra is a noisy system, yet since death only occurs when resources become limited, energy is shared more liberally, and interactions are not limited

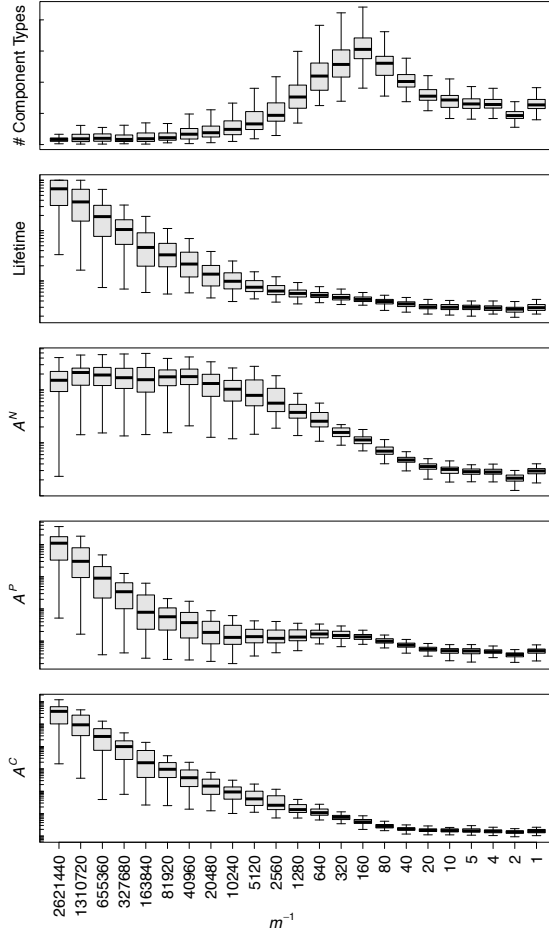


Figure 4: Effect of change in mutation rate in Stringmol between the range $m = 2621440^{-1}$ and $m = 1$ mutations per base copied. From top to bottom: Number of species produced in a run; A^N , A^P and A^C .

by any ability to bind, there is activity even when mutation goes beyond the rate at which mutation causes “error catastrophe” (Eigen, 2002). The major drawback of Stringmol is that the emergence of parasites appears to be guaranteed. The specificity of the binding routine appears to be a limiting factor in the ability of the system to explore the fitness landscape before perishing. However, it is worth noting that the pathway of mutation between opcodes in the configuration of Stringmol we used here has already been improved upon (Droop and Hickinbotham, 2011b). The new measure we have devised here will be a useful additional tool in designing a more benign mutation configuration.

Conclusion

Whilst established measures of evolutionary activity based on presence or counts of components yield useful visualisations of evolutionary data, they do not give a quantitative

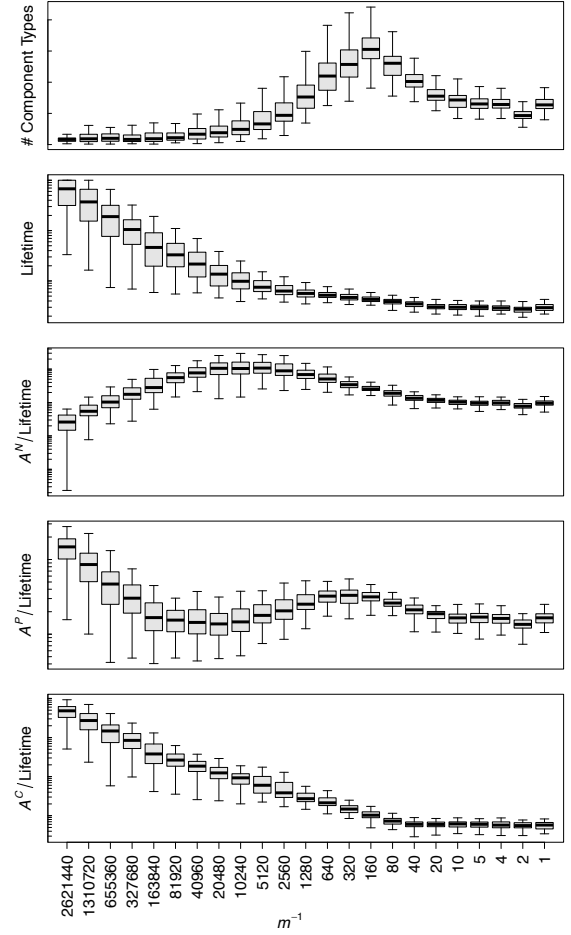


Figure 5: Effect of change in mutation rate in Stringmol between the range $m = 2621440^{-1}$ and $m = 1$ per base copied *divided by the life time of the trial*. From top to bottom: Number of species produced in a run; Lifetime of a run; A^N ; A^P ; and A^C .

measure that can be used for tuning the system, especially where fitness in the system is intrinsic. We have presented a new measure of evolutionary activity that satisfies this requirement. We have evaluated the approach on two systems that we understand sufficiently to interpret the results. The predicted dynamics were observed experimentally, and successfully detected by the new EA measure.

The dynamics of Stringmol showed similar, if more restrained effects to those exhibited by Tierra. The most marked difference between the two platforms occurred at high mutation rates, where the system was incapable of surviving beyond the initial population. Utilising the new EA measure, we plan to investigate how design of the seed component, the binding strategy and the way opcodes are interchanged through mutation can change the ability of the system to produce open-ended novelty.

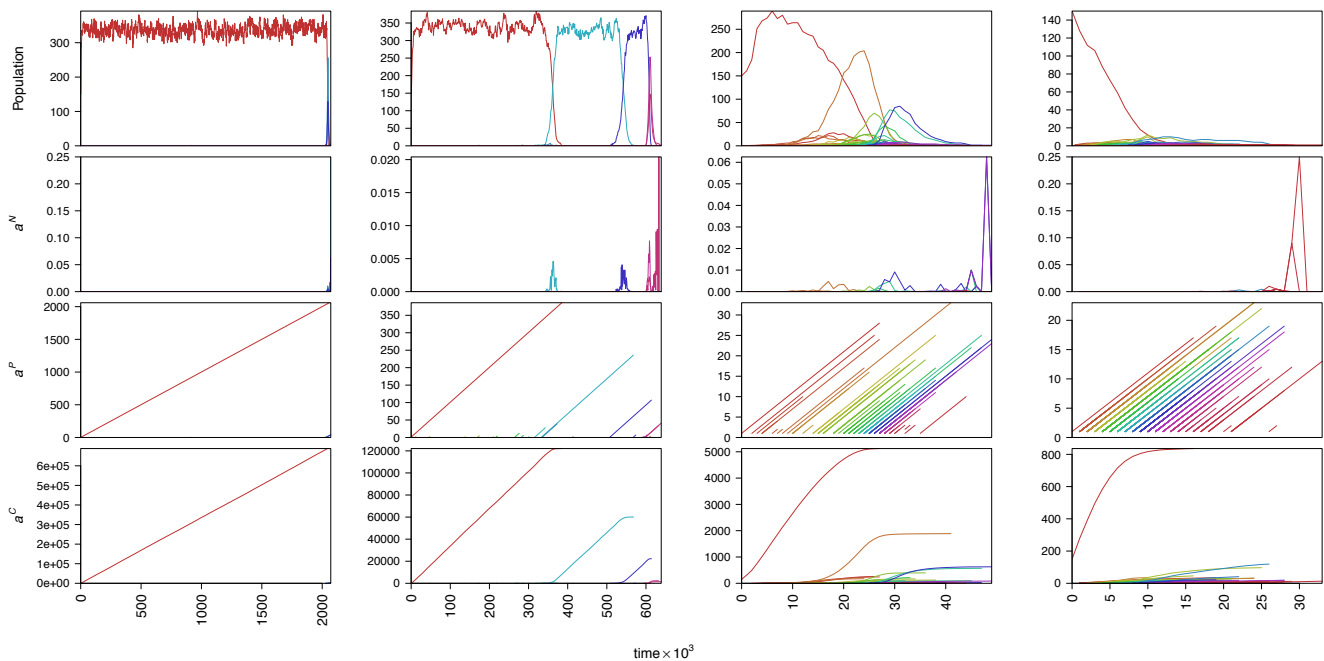


Figure 6: Illustrations of median-scoring runs from the set of 100 trials of Stringmol at values of m^{-1} of (from left to right) 2,622,440; 163,840; 5,120; 160. A^N was maximal at $m^{-1} = 163,840$ (centre-left column), whereas $A^N/\text{Lifetime}$ was maximal at $m^{-1} = 5,120$ (centre-right column).

In real biology all fitness is intrinsic as there is no fitness function. As ALife systems become increasingly sophisticated, intrinsic fitness is bound. Our approach focusses on minimising the contribution of neutral mutations to the dynamics of an evolving system, which allows intrinsic fitness of individuals to be revealed as they become active in establishing a component in a population.

Acknowledgements

SJH was supported by EPSRC. APD is funded by the CRUK.

References

- Bedau, M. A., Snyder, E., Brown, C. T., and Packard, N. H. (1997). A comparison of evolutionary activity in artificial evolving systems and in the biosphere. In *ALife IV*, pages 125–134. MIT Press.
- Bullock, S. and Bedau, M. A. (2006). Exploring the dynamics of adaptation with evolutionary activity plots. *Artificial Life*, 12:193–197.
- Channon, A. (2006). Unbounded evolutionary dynamics in a system of agents that actively process and transform their environment. *Genetic Programming and Evolvable Machines*, 7:253–281.
- Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial chemistries – a review. *Artificial Life*, 7(3):225–275.
- Droop, A. and Hickinbotham, S. (2011a). Application of small-world mutation topologies to an artificial life system. In *ECAL 2011*, pages 208–209. MIT Press.
- Droop, A. and Hickinbotham, S. (2011b). Properties of biological mutation networks and their implications for alife. *Artificial Life*, 17:353–364.
- Droop, A. and Hickinbotham, S. (2011c). Rstringmol software. <http://www-users.cs.york.ac.uk/~sjh/software/>.
- Eigen, M. (2002). Error catastrophe and antiviral strategy. *Proceedings of the National Academy of Sciences of the United States of America*, 99(21):13374–13376.
- Hickinbotham, S., Clark, E., Stepney, S., Clarke, T., Nellis, A., Pay, M., and Young, P. (2010a). Diversity from a monoculture: Effects of mutation-on-copy in a string-based artificial chemistry. In *ALife XII*. MIT Press.
- Hickinbotham, S., Clark, E., Stepney, S., Clarke, T., Nellis, A., Pay, M., and Young, P. (2010b). Specification of the stringmol chemical programming language version 0.1. Technical Report YCS-2010-458, Univ. of York.
- Rav, M. (2011). Tierra by Tom Ray. <http://tierra.lolwh.at/>.
- Ray, T. (1991). An approach to the synthesis of life. In Langton, C., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II*, volume XI of *Santa Fe Institute Studies in the Sciences of Complexity*, pages 371–408. Addison-Wesley.
- Ray, T. and Xu, C. (2001). Measures of evolvability in tierra. *Artificial Life and Robotics*, 5(4):211–214.

Robustness and evolvability of cooperation

Antoine Frénoy¹, François Taddei¹ and Dusan Misevic¹

¹ Center for Research and Interdisciplinarity, INSERM U1001, Université Paris Descartes, Sorbonne Paris Cité
frenoy@gmx.com

Abstract

Robustness and evolvability are indirectly selected properties of biological systems that still play a significant role in determining evolutionary trajectories. Understanding such second order evolution is even more challenging when considering traits related to cooperation, as the evolution of cooperation itself is governed by indirect selection. To examine the robustness and evolvability of cooperation, we used an agent-based model of digital evolution, Aevol. In Aevol individuals capable of cooperating via costly public good secretion evolve for thousands of generations in a classical tragedy of the commons scenario. We varied the cost of secreting the public good molecule between and within individual experiments and constructed and evaluated millions of mutants to quantify the organisms' position in the fitness landscape. Populations initially evolved at different regimes selecting against secretion, and then continued the evolution at a reasonably low cost of secretion. The populations that experienced a very strong selection against cooperation evolved less secretion than the ones that initially experienced a less drastic selection against cooperation via a high secretion cost. The mutational analysis revealed a correlation between the number of mutants with increased secretion and the secretion level across all costs of secretion. We also evolved several clones of each population to highlight a strong effect of history in general on cooperation. Our work shows that the history of cooperative interactions has an effect on evolutionary dynamics, a result likely to be relevant in any cooperative systems that are frequently experiencing changes in cost and benefit of cooperation.

Introduction

The interplay between robustness and evolvability is one of the central questions in evolutionary biology (Wagner, 2005; Lenski et al., 2006). While mutation robustness should be beneficial, due to avoiding deleterious mutations and maintaining the organism's phenotype, without the ability to adapt to a novel environment the organism may perish in a changing world. Both selection for robustness and for evolvability are indirect, making these properties potentially difficult to investigate experimentally. Past research has found evidence that evolvability (Bedau and Packard, 2003; Earl and Deem, 2004; Wagner and Altenberg, 1996; Woods et al., 2003), as well as robustness can be selected for (Altenberg,

2005; Wilke et al., 2001; Misevic et al., 2006; Azevedo et al., 2006) under a range of circumstances. However, in most of these studies the traits that evolved different robustness and evolvability had direct fitness benefit and were thus under direct selection. We extend this work by studying aspects of evolvability and robustness of an indirectly selected trait, specifically cooperation via public good secretion.

Cooperation among individuals is frequently present in natural world and yet it remains a fascinating evolutionary enigma. When helping others comes at a direct personal cost, natural selection predicts that individuals who do not cooperate would be favored over cooperating ones. A number of theories exist to explain the diversity and abundance of stable cooperation systems in nature, primarily relying on inclusive fitness, kin and group selection arguments (Axelrod, 1984; Sober and Wilson, 1998; Lenski et al., 2006; Nowak, 2006; Lehmann and Keller, 2006; Lehmann et al., 2007). Public good secretion in microbes has been a particularly successful model system for the study of the evolution of cooperation, allowing for great insight into the forces that shape its emergence and persistence (West et al., 2007; Racey et al., 2010).

The majority of both theoretical and experimental work on robustness and evolvability has been done under either fixed environmental conditions or traits that have direct fitness effects. Here we study cooperation, a trait under indirect selection, during evolution in variable environment, where the fitness cost of cooperation changes. To investigate the effect of evolutionary history in general, and changing costs of cooperation in particular, on the evolution of cooperation, we use a digital evolution platform, Aevol. As in bacteria, the public good in Aevol is a molecule that is secreted into the environment at a cost and can then benefit both the producer and all its neighbors, acting as an agent of cooperation. After establishing the parameter range allowing for the appearance of secretion, we performed experiments investigating whether strong selection against secretion will lead to genotypes residing in regions of the fitness landscape far away from cooperation. In other words, we wanted to test the hypothesis of strong selection against se-

cretion not only causing a direct pressure against secretion genes, but also an indirect pressure on the genome structure that will modify the generic architecture and make secretion genes less likely to appear via mutations. In nature, cooperative phenotype may have to repeatedly evolve after being outcompeted by “cheaters”, organisms benefiting from the cooperation without contributing to it. Depending on phenotype frequencies and ecological interactions between different types of individuals present, the cost to benefit ratio of cooperation would change. Understanding these history effects is necessary for understanding the long-term evolution of cooperation and may also be relevant to treatment of bacterial infections whose pathogenicity depends on cooperation among individuals, such as *Pseudomonas aeruginosa* (Griffin et al., 2004).

Methods

Description of the model system

In this study we use the Aevol platform (Knibbe et al., 2008; Parsons, 2011), an individual-based, genetic algorithm-inspired model aimed at studying the evolutionary processes. It is especially well suited for examining the indirect selection pressures on the genome structure due to microbial-inspired, complex genotype-phenotype map (Parsons et al., 2010). The genomic layer of Aevol is inspired by bacterial genomic, but should be general enough for our needs. Aevol is an open-source project and is freely available at www.aevol.fr/download. In all our experiments we used the default parameters unless otherwise noted.

The genome of Aevol individuals is encoded by a double-stranded string of zeros and ones. The phenotype is a collection of traits that are represented by a 2D curve, each point on the curve specifying performance level for an abstract biological process, a metabolic trait. A single protein is obtained by transcription and translation of the binary genome strings, through a mathematical transformation. To be expressed, protein sequence must be found between start and stop codons, that in turn must be between a promoter and terminator sequences, and be preceded by a Shine-Dalgarno sequence. A protein can affect a number of different processes simultaneously, to a different degree, depending on its expression level. There is no explicit genetic regulation in this version of Aevol, but there are functional interactions (combining the effect of two proteins contributing to the same trait). The transcription efficiency, and thus the protein expression level can be affected by mutations in the promoter region. Such genotype-to-phenotype map is directly inspired by the complexities of bacterial genomics and allows us to study not only the evolutionary dynamics of phenotypic traits, but also the evolution of the genetic architecture supporting these traits, including the genome length, percentage of coding/non-coding DNA, number of

genes, and number of operons (Knibbe et al., 2007b,a; Parsons, 2011; Parsons et al., 2010).

The fitness of an Aevol digital organism is a decreasing function of the gap between the curve representing its phenotype and a target curve representing the “perfect phenotype” for the chosen environment. This target phenotype is a combination of several gaussians, chosen by the researcher and fixed during the experiments. There may be many ways to encode the same protein and thus many genotypes may map to the same phenotype. Moreover, different phenotypes may have the same fitness. In our system, selection acts on the phenotypic variation created by random mutations of organisms’ genome. We distinguish between two types of mutations: small mutations (single base substitutions, insertion or deletion of up to 6 neighboring bases) and large mutations (duplication, deletion, inversion, or translocation of a section of the genome whose size and location are chosen at random). The mutation rates we used are 5×10^{-5} per nucleotide per generation for small mutations, and 5×10^{-6} for large mutations. Given the typical genome size of 10^4 bases, for each individual we expect about one small mutation per generation and one large mutation every 5 generations. The stochastic nature of our model is derived from the random choice of mutations at each generation, combined with the probabilistic selection which we describe below. By modifying the random number seed, we can perform multiple experiments with the same set of parameters and analyze the statistical significance of our results.

In order to study robustness and evolvability of cooperation we extended the Aevol system to include the possibility of secreting and consuming a public good, a diffusible, degradable molecule that is produced at a cost but confers a benefit to each individual absorbing it (West et al., 2007; Racey et al., 2010). Based on the studies of public good dynamics in Aevol and other systems (Brown and Taddei, 2007; Misevic et al., 2012), we set the degradation rate to 10% per generation (the amount of the public good molecule that degrades each generation) and diffusion rate to 5% (the percentage of the public good that diffuses into each of the neighboring cells in the classical 3x3 Moore neighborhood). Under this scenario, 54% of the initially present public good remains in the grid cell after each generation.

To allow for the encoding of the public good production, we modified the genotype-phenotype map as follows: half of the phenotypic traits remain related to the “classical” metabolic phenotype and their levels have a direct effect on fitness, while the other half specifies the secretion-related phenotype. The metabolic fitness component is inversely proportionate to the gap between the metabolic part of the phenotype and the target phenotype. The gap between the secretion part of the phenotype and the secretion target phenotype is inversely proportionate to the amount of public good secreted by an individual. The total fitness of an organism is the combination of its metabolic fitness, the cost

it pays for secreting the public good and the benefit it gets from any public good present in its local environment. To be precise, $W = W_{met} * (1 + B * (PG - C * S))$, where W is the total fitness, W_{met} is the metabolic fitness, PG is the amount of public good present in the local environment, S is the amount of the public good secreted by the individual, B is the contribution of cooperation to fitness (set to 0.5 in all our experiments), and C is the cost of secretion that we will vary in some of our experiments. As an individual does not directly benefit from the public good it secretes, but only from the public good secreted by its ancestors and neighbors, the selection for cooperation is indirect.

Spatial structure is thought to have a major impact on the evolution of public-good secretion: cooperation is likely to be favored by kin selection when related individuals are spatially close to each other (West et al., 2007; Nowak and May, 1992; Hauert and Doebeli, 2004). In order to enable the potential evolution of cooperation, our individuals evolve in a square toroidal grid with 1024 positions (32x32). Each position is inhabited by a single individual and there are no empty positions. The selection is done on a purely local basis: to compute a new generation, for each grid position we synchronously compete the nine individuals in its neighborhood. The higher the fitness of an individual is, the higher is the probability it will reproduce. All mutations happen during the reproduction step, after which the fitness of the new individual is recomputed, based on the changed levels of the available public good and mutations that occurred. To avoid the drastic decrease of the selection pressure as organisms approach the target phenotype, we use rank based, rather than fitness based selection in the neighborhoods. Additionally, the rank contributes exponentially to the probability of being selected for reproduction, in line with previous work on genetic algorithms in general and Aevol in particular (Bickle and Thiele, 1994; Knibbe et al., 2007b). We choose the exponential rank selection parameters that give the individual with the highest fitness in the neighborhood a 31.3% probability of reproducing in the central cell of that neighborhood, while that probability is 1.8% for the individual with the lowest fitness. We determined these selection probabilities by testing a range of parameters and choosing ones that result in evolution of the highest level of secretion over time (data not shown).

Experimental design

Secretion cost and the evolution of cooperation. The ratio between the cost paid by the individual that produces the public good and the benefit received from its consumption is a crucial parameter affecting the evolution of cooperation (Hamilton, 1964; Nowak, 2006; West et al., 2007). In order to quantify the dynamics of cooperation in Aevol under different cost-benefit ratios, we performed 50 experiments for each of the 7 different levels of secretion cost, $C = 0.01, 0.05, 0.1, 0.2, 0.3, 1$ and 2 . Each experiment lasted 30,000

generations and we recorded the average amount of the public good secreted by the individuals over time. We used the results from these experiments to inform our parameter choices in remainder of the study.

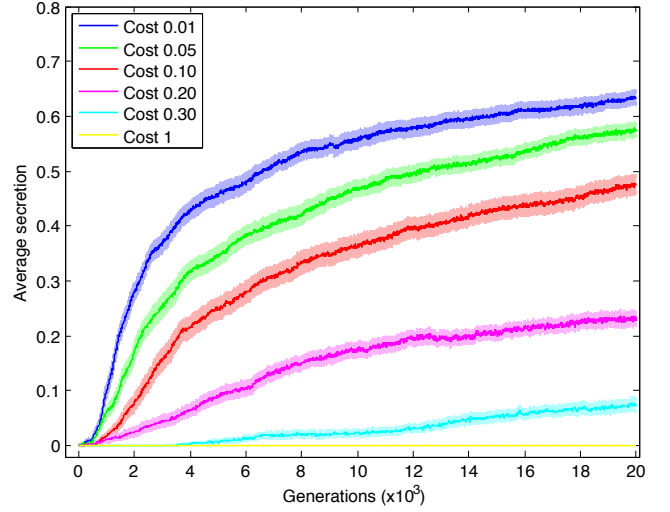


Figure 1: Effect of secretion cost on the evolution of cooperation. Each line represents an average of 50 replicate experiments conducted at the same secretion cost. The shaded area is one standard error of the mean. Results for cost = 2 are indistinguishable from cost = 1 and are thus not shown.

Historical cost of secretion and the evolution of cooperation. To quantify the strength of the historical effects, as well as robustness and evolvability of cooperation in Aevol, we performed a series of experiments in which populations evolved for 10,000 generations at one of the three regimes with different cost of secretion, specifically $C = 0.8, C = 0.5, C = 0.35$. We also tested an additional regime, *NoSec*, where the biological processes that were assigned to the secretion part of the phenotype are associated with metabolism instead and their optimal expression level is set to zero. The cost parameters we chose should completely inhibit the evolution of cooperation, or allow for it only at extremely low levels. After 10,000 generations the cost of secretion is set to $C = 0.25$ for all treatments, and the secretion target phenotype in *NoSec* treatment becomes the same as in the three other treatments. Specifically, the values y for all processes in the target phenotype with $x \in (0, 1)$ are described by four Gaussian functions of the form $y = H e^{-(x-M)^2/2W^2}$, where $(H, M, W) = \{(0.35, 0.3, 0.04), (0.5, 0.2, 0.02), (0.5, 0.7, 0.02), (0.35, 0.8, 0.04)\}$. All processes with x -values less than 0.5 are associated with metabolism while the others are associated with secretion. During all these experiments we recorded the average amount of secreted compound.

Mutational robustness. We analyzed the genetic architecture of all the individuals from each population at generation 10,000 by performing large number of mutations and recording the overall fitness and the amount of the public good secreted by the mutants. Each organism was reproduced 10,000 times with its offspring having the probability of acquiring mutations in the same way as during the reproduction in typical experiments, for a total of 10,240,000 mutants analyzed from each population. We evaluated the frequency of beneficial, neutral and deleterious mutations as well as their magnitude.

History versus chance. To quantify the effect of history (versus chance) on the amount of secretion after generation 10,000, we performed an experiment similar to the classic “adaptation, chance and history” studies (Travisano et al., 1995; Wagenaar and Adami, 2004). In these experiments, for each of our cost treatments, we selected 10 populations at random as the available computational power did not allow us to study all 100 populations per treatments. Each of these 10 populations was cloned 10 times when releasing the secretion cost (generation 10,000), to obtain 10 groups of 10 replicates. We measured the average amount of secreted compound during 3,000 additional generations for each of these populations. An analysis of the variance between the different groups compared to the variance within each group provides a measure of the influence of history and chance on the evolution of these populations. We do not specifically discuss the effects of adaptation here as they are apparent from the change in amount of cooperation in all treatments.

Results and discussion

Direct relationship between evolved secretion and its cost. The cost of secreting the public good had a direct and strong effect on the average amount of secreted public good molecule (Fig. 1). This is in accordance with our expectations, both in terms of the direct trade-off between cost and benefit of cooperation and in relation to classical results (West et al., 2007; Nowak, 2006). We used these experiments to establish a baseline cost of cooperation for which no population would evolve and maintain significant levels of secretion during at least the initial 10,000 generations of evolution. In particular, we find that costs higher than 0.3 have this property and are thus suitable for use in the experiments from the second part of our study.

History affects future secretion levels. The phenotype of individuals that evolved for 10,000 generations under high costs of secretion or *NoSec* regime was generally identical: they did not secrete any public good molecules, as expected. However, once the selection pressure against secretion was released (at generation 10,000), the fates of different populations quickly diverged. By 10,000 generations, mutations and evolution erased any statistical differences between the

treatments so we used an earlier time point in our analysis. Rather than using just the final secretion which may be strongly affected by stochastic factors, we measured the amount of cooperation that evolved by averaging the amount secreted during the first 3,000 generations after releasing secretion cost (Fig. 2), and used the Mann-Whitney non-parametric test to compare different treatments. We find a general trend of lower secretion in populations that underwent the *NoSec* regime (strong direct selection against secretion) compared to the ones that experienced a high cost of secretion (less drastic selection against secretion) in their past (Mann-Whitney U test, $p = 0.010$). However we did not find any significant difference between the three secretion costs. This trend, although very noisy at our levels of replication, indicates that genotypes have preserved some information of their evolutionary history. The ones that evolved with strong direct pressure against secretion (*NoSec* treatment) are more robust and less likely to change, while the ones that evolved with less strong pressure via secretion cost are more evolvable.

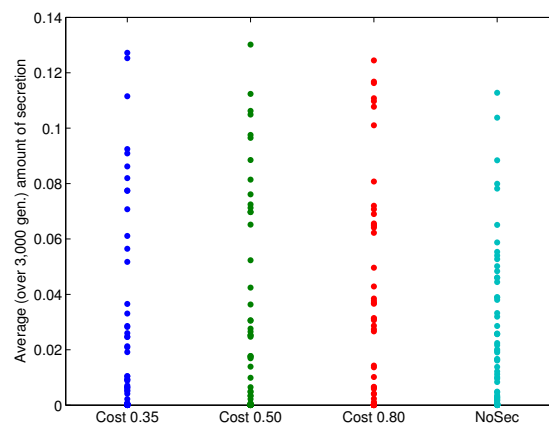


Figure 2: Average amount of secretion between generation 10,000 and generation 13,000, sorted by the cost regime in the first 10,000 generations. Each point represents a single replicate population. There are 100 independent replicates for each treatment.

Mutational robustness is strongly correlated with future secretion levels. Specifically, we suspect that the genotypes that evolved robustness against secretion were located in regions of the fitness landscape mutationally far away from genotypes that confer the secretion phenotype. To test for such genotypic memory, we performed a mutagenesis test (Fig. 3), as described in the methods. We found a strongly significant difference in the proportion of mutants with increase in secretion (weighted by the magnitude of these effects) between on one side the three high cost treat-

ments and on the other side the *NoSec* treatment (Welch's t-test, $p = 0.0001$), but no significant difference when comparing the three different costs between them. We furthermore found a very strong within-treatment correlation between the proportion of mutants with increase in secretion (weighted by the magnitude of these effects) and the average amount of public good secreted during the first 3,000 generations after regime change (Table 1). This correlation is still present if we pool all the data, with the coefficient of correlation of 0.37 and $p < 10^{-13}$, and suggests that history, encoded as genotypic memory, does strongly matter.

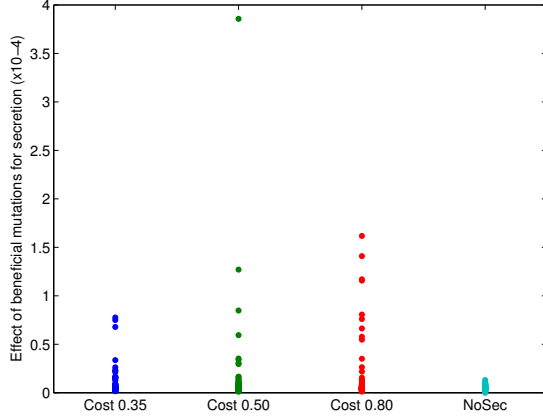


Figure 3: Beneficial mutations for secretion at generation 10,000, depending on the regime during the first 10,000 generations. Each point represents the average effect of 10,240,000 mutations within single replicate population. There are 100 independent replicates for each treatment.

Cost of secretion	Correlation coefficient	p-value
0.35	0.6639	$< 10^{-13}$
0.5	0.3669	$< 10^{-3}$
0.8	0.4134	$< 10^{-4}$
NoSec	0.1790	0.07

Table 1: Correlation between the proportion of mutants with increase in secretion (weighted by the magnitude of these effects) at generation 10,000 and the average amount secreted between generation 10,000 and generation 13,000 for each treatment.

ANOVA shows a strong effect of history versus chance. Following the experimental protocol described in the methods, we performed a one-way ANOVA to assess the influence of history (versus chance) on the evolution of secretion (Table 2). We found a significant influence of history for each of our three cost treatments, even if this history is not

necessarily dependent on the cost of secretion, as we expected initially. As our previous experiments already found the *NoSec* treatment to have different historical effect on robustness and evolvability of the cooperation phenotype, here we omitted it from the analysis and focused instead on the historical effect of the three cost treatments.

Cost of secretion	SShist/SStot	F statistic	p-value
0.35	0.63	16.9349	$< 10^{-15}$
0.5	0.44	7.7311	$< 10^{-7}$
0.8	0.57	13.3905	$< 10^{-13}$

Table 2: Influence of history versus chance on secretion. SShist is the sum of squares due to history, while SStot is the total sum of squares (history plus chance).

Conclusion

Using the Aevol digital system we performed a series of experiments to test the effect of evolutionary history on the robustness and evolvability of cooperation. Our results generally showed a weak effect of the strength of selection against secretion on the future evolution of secretion, and a strong effect of history in general. The data was extremely noisy and may require a much greater number of replicates than we could produce for this study. The difference in the mutational neighborhood occupied by populations that have evolved at different secretion costs was not significant; however, the difference between the three cost-driven regimes (indirect pressure against secretion due to moderately high cost) and the NoSec regime (strong direct pressure against secretion) was large. Moreover the accessibility of beneficial mutations for secretion did strongly correlate with the amount of secretion in our experiments, generally validating the mutational analysis approach. The analysis of several clones of each population highlighted a strong influence of history on the robustness and evolvability of cooperation, however the cost of cooperation does not seem to be the main factor creating this history. Much research remains to be done in terms of fully understanding these complex interactions.

Acknowledgements

We thank the computation center of the national institute of nuclear and particle physics (IN2P3) of the CNRS for providing the computational power needed by this study and the four anonymous reviewers for their helpful and frank comments.

Contributions

AF conceived and designed the study, performed the experiments, analyzed the results, and wrote the paper. FT provided input on the study design, result analysis and interpre-

tation. DM wrote and edited the paper and contributed to data analysis and presentation.

References

- Altenberg, L. (2005). Evolvability suppression to stabilize far-sighted adaptations. *Artificial life*, 11:427–443.
- Axelrod, R. (1984). *The Evolution of Cooperation*. Basic Books, New York, NY.
- Azevedo, R. B. R., Lohaus, R., Srinivasan, S., Dang, K. K., and Burch, C. L. (2006). Sexual reproduction selects for robustness and negative epistasis in artificial gene networks. *Nature*, 440:87–90.
- Bedau, M. and Packard, N. (2003). Evolution of evolvability via adaptation of mutation rates. *BioSystems*, 69:143–162.
- Bickle, T. and Thiele, L. (1994). A comparison of selection schemes used in genetic algorithms. *Evolutionary Computation*, 4:361–394.
- Brown, S. and Taddei, F. (2007). The durability of public goods changes the dynamics and nature of social dilemmas. *PLoS ONE*, 2:e593.
- Earl, D. and Deem, M. (2004). Evolvability is a selectable trait. *Proceedings of the National Academy of Science USA*, 101:11531–11536.
- Griffin, A. S., West, S. A., and Buckling, A. (2004). Cooperation and competition in pathogenic bacteria. *Nature*, 430:1024–1027.
- Hamilton, W. D. (1964). The genetical evolution of social behavior i+ii. *Journal of Theoretical Biology*, 7:1–52.
- Hauert, C. and Doebeli, M. (2004). Spatial structure often inhibits the evolution of cooperation in the snowdrift game. *Nature*, 428:643–646.
- Knibbe, C., Coulon, A., Mazet, O., Fayard, J., and Beslon, G. (2007a). A long-term evolutionary pressure on the amount of noncoding dna. *Molecular Biology and Evolution*, 24:2344–2353.
- Knibbe, C., Fayard, J.-M., and Beslon, G. (2008). The topology of the protein network influences the dynamics of gene order: From systems biology to a systemic understanding of evolution. *Artificial Life*, 14:149–156.
- Knibbe, C., Mazet, O., Chaudrier, F., Fayard, J., and Beslon, G. (2007b). Evolutionary coupling between the deleteriousness of gene mutations and the amount of non-coding sequences. *Journal of Theoretical Biology*, 244:621–630.
- Lehmann, L. and Keller, L. (2006). The evolution of cooperation and altruism: A general framework and a classification of models. *Journal of Evolutionary Biology*, 19:1365–1376.
- Lehmann, L., Keller, L., West, S., and Roze, D. (2007). Group selection and kin selection: Two concepts but one process. *Proceeding of the National Academy of Sciences USA*, 104:6736–6739.
- Lenski, R. E., Barrick, E. J., and Ofria, C. (2006). Balancing robustness and evolvability. *PLoS Biology*, 4:e428.
- Misevic, D., Frenoy, A., Parsons, D., and Taddei, F. (2012). Effects of public good properties on the evolution of cooperation. *Artificial Life XIII, Proceedings of the Thirteenth International Conference on Synthesis and Simulation of Living Systems*.
- Misevic, D., Ofria, C., and Lenski, R. E. (2006). Sexual reproduction reshapes the genetic architecture of digital organisms. *Proceedings of the Royal Society of London Series B-Biological Sciences*, 273:457–464.
- Nowak, M. (2006). Five rules for the evolution of cooperation. *Science*, 314:1560–1563.
- Nowak, M. A. and May, R. M. (1992). Evolutionary games and spatial chaos. *Nature*, 359:826–829.
- Parsons, D. (2011). *Indirect Selection in Darwinian Evolution: Mechanisms and Implications*. PhD thesis, INSA.
- Parsons, D. P., Knibbe, C., and Beslon, G. (2010). Importance of the rearrangement rates on the organization of transcription. In Fellermann, H., Drr, M., Hanczyc, M. M., Ladegaard Laursen, L., Maurer, S., Merkle, D., Monnard, P.-A., Stoy, K., and Rasmussen, S., editors, *ALife XII*, pages 479–486. MIT Press, Cambridge, MA.
- Racey, D., Inglis, R. F., Harrison, F., Oliver, A., and Buckling, A. (2010). The effect of elevated mutation rates on the evolution of cooperation and virulence of *Pseudomonas aeruginosa*. *Evolution*, 64:515–521.
- Sober, E. and Wilson, D. S. (1998). *Unto Others: The Evolution and Physiology of Unselfish Behavior*. Harvard University Press, Cambridge, MA.
- Travisano, M., Mongold, J., Bennett, A., and Lenski, R. (1995). Experimental tests of the roles of adaptation, chance, and history in evolution. *Science*, 267:8790.
- Wagenaar, D. and Adami, C. (2004). Influence of chance, history, and adaptation on digital evolution. *Artificial Life*, 10:181–190.
- Wagner, A. (2005). *Robustness and Evolvability in Living Systems*. Princeton University Press, Princeton, NJ.
- Wagner, G. and Altenberg, L. (1996). Complex adaptations and the evolution of evolvability. *Evolution*, 50:967–976.
- West, S. a., Diggle, S. P., Buckling, A., Gardner, A., and Griffin, A. S. (2007). The social lives of microbes. *Annual Review of Ecology, Evolution, and Systematics*, 38:53–77.
- Wilke, C., Lan Wank, J., Ofria, C., Lenski, R., and Adami, C. (2001). Evolution of digital organisms at high mutation rates leads to the survival of the flattest. *Nature*, 412:331–333.
- Woods, R., Barrick, J., Cooper, T., Shrestha, U., Kauth, M., and Lenski, R. (2003). Second-order selection for evolvability in a large escherichia coli population. *Science*, 331:1433–1436.

Evolution of Self-Replicating Cube Conglomerations in a Simulated 3D Environment

Paul Grouchy^{1,2} and Hod Lipson²

¹Institute for Aerospace Studies, University of Toronto, Toronto, ON, Canada M3H 5T6

²Mechanical & Aerospace Engineering, Cornell University, Ithaca, NY, USA 14850
paul.grouchy@utoronto.ca, hod.lipson@cornell.edu

Abstract

The evolution of self-replication in three dimensions is explored for the first time. A discrete three-dimensional world populated with physically-realizable “molecubes” is simulated. The cubes have randomly initialized controllers, can rotate about an axis, and can attach to one another to form conglomerations. Genetic material, which defines cube controllers, is exchanged stochastically between attached cubes and subject to random mutations. Self-replicating cube conglomerations emerge in this simulation across a wide range of densities and without the use of a fitness function, yielding insight into the evolution of self-replication in nature and furthering progress toward physically-realizable self-replicating machines.

Introduction

Researchers have been interested in artificial life simulations for as long as digital computers have existed. Early on, von Neumann invented cellular automata [Neumann (1966)], which are still an active area of research to this day. While the original cellular automata were programmed with the ability to self-replicate, more recent experiments have demonstrated the spontaneous emergence of replicators in such systems [Chou (1997)].

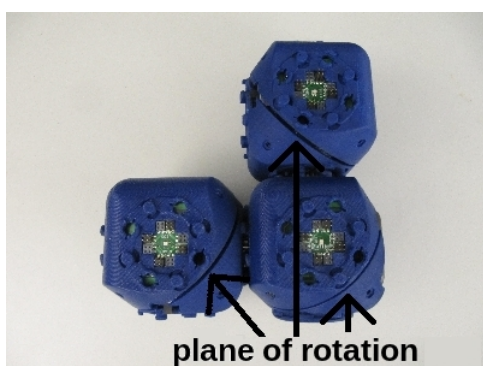


Figure 1: Three physical “molecubes”. Note the plane of rotation.

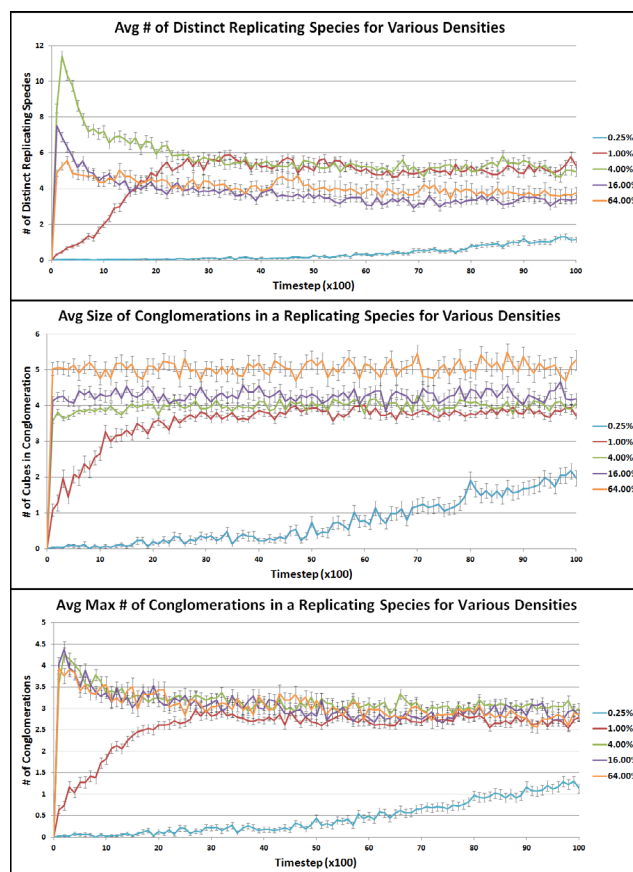


Figure 2: Experimental results for various densities. A replicating species is defined as a genome that occurs in two or more genetically homogeneous molecube conglomerations, where each conglomeration contains at least two cubes. Each result is an average over 100 randomly initialized runs and error bars show standard error.

In cellular automata simulations, every agent is identical (i.e. they all use the same ruleset). More complex artificial life paradigms such as Tierra [Ray (1992)] and Avida [Adami and Brown (1994)] simulate a diverse population of

digital organisms that compete for computational resources, which can then be used for replication. Each agent in these simulations contains its own instruction set or “program” that can evolve over time. Organisms in *Avida* have the ability to self-replicate by running instructions to allocate memory for a child program and copy their instruction set into this memory. There is no explicit fitness function guiding evolution in these simulations, allowing for comparisons to self-replicating life on Earth. While further analogies can be drawn between these computational programs and real-world systems, it is difficult to imagine physical implementations of these artificial life-forms.

In an effort to narrow the gap between computational simulation and the physical world, a 2D simulation of non-uniform cellular automata that were physically realizable was designed and run in [Studer and Lipson (2005)]. The automata instruction sets existed in simulated “molecubes,” which are cubes that can attach to one another using electromagnets and can rotate their halves around a fixed axis (see Figure 1). Physical versions of these cubes have previously been constructed and [Zykov et al. (2005)] demonstrated how a group of these molecubes could construct an identical second group using other molecubes. Preliminary results from the 2D simulations demonstrated, without the use of a fitness function, spontaneous emergence of self-replication. A group of simulated molecubes with identical rulesets (a “species”) collected other molecubes in the 2D environment, transferred their rulesets, and then separated into two identical molecube groupings. A variety of self-replicating species often co-existed simultaneously, competing for molecube resources in the simulation.

The experiments presented in this paper bring ALife another step closer to realizable real-world systems by demonstrating the spontaneous emergence of self-replication in a population of physically realizable three-dimensional molecubes that exist in a simulated three-dimensional world. While this environment lacks several properties of the physical world, most notably gravity, this is the first time that the emergence of self-replication has been observed in three dimensions. Replicators emerged in simulations of varying densities, producing examples of agents that must move through the environment to accumulate cubes as well as replicators that were forced to remain largely stationary. This mirrors the independent rise of multicellularity in plants and animals [Bonner (1998)].

3D Physical Cube Automata

The simulated cubes in the following experiments were based on real “molecubes,” presented in [Mytilinaios et al. (2004)]. Each of these physical cubes contains an actuator that allows it to rotate one of its pyramid-shaped halves in 120° increments and adjacent cubes can connect to one another using electromagnets. Adjacent cubes can also communicate over a digital channel. Figure 1 shows an example

of these physical molecubes.

The computer simulations consisted of a population of simulated molecubes that exist in a three-dimensional $N \times N \times N$ environment partitioned into a 3D grid. Each discrete grid location can either be vacant or occupied by a molecube. A single molecube cannot move from one discrete location to another, however a molecube can move other molecubes that are attached to it by rotating around its axis. One can then imagine various methods of locomotion whereby attached molecubes take turns rotating around their respective axis. Gravity is not incorporated into the simulation, therefore groups of molecubes can move in any direction. The simulated world wraps around, i.e. it is toroidal. If a molecube rotation creates a collision (i.e. two molecubes occupying the same 3D grid location), this move is reversed. To reduce the computational complexity of the system, collisions *during* a molecube rotation are ignored. Furthermore, a maximum of 15 molecubes could be attached together in a single group, and loops of attached molecubes were not allowed.

Each simulated molecube contains a controller that updates the cube’s output set y based on its previous outputs and its current input values x . See Table 1 for descriptions of the controller inputs and outputs. During a simulation, each molecube’s controller is evaluated once per timestep. The order in which the controllers are evaluated is based on inter-molecube connections. Therefore while it is not random, it does vary over time.

The controllers used are 0D3v0 controllers [Grouchy and D’Eleuterio (2010)], where there is one evolvable ordinary differential equation per controller output y_n (see Equation 1).

$$dy/dt = f(x, y) \quad (1)$$

The functions f_n are represented as trees and can incorporate constants, inputs, outputs and a variety of mathematical operations (as in symbolic regression in Genetic Programming [Poli et al. (2008)]). For details on how the controllers are initialized, evaluated, and mutated, the reader is referred to [Grouchy and D’Eleuterio (2010)]. Crossover at the function level was not implemented for our experiments, however tree-level crossover that overwrites a randomly selected subtree with a randomly selected subtree from another controller was used. When at least one cube is attached to a cube selected for mutation, tree-level crossover is performed *instead of a mutation* with a probability of 0.5.

At each timestep, there is a probability μ that a random mutation will occur within a molecube’s genome. Furthermore, if a molecube is attached to at least one other cube, there is a 50% chance that it will have its ODEs overwritten by an attached neighbour’s ODEs. This can occur once per attached cube, per timestep. By stochastically deciding whether a cube’s equations are to be overwritten by a

Parameter	Range	Description
$x_n, n \in [0, 5]$	$\{0,1\}$	Incoming communication bit from molecule adjacent to side n (0 if no adjacent cube). Note that molecules do not have to be attached to communicate.
$x_n, n \in [6, 11]$	$\{0,0.5,1\}$	Adjacent/attached inputs. Set to 0 if no molecule adjacent to side $n - 6$, 0.5 if molecule adjacent but not attached, 1 if molecule adjacent and attached.
$y_n, n \in [0, 5]$	$[0,1]$	Outgoing communication bit to molecule adjacent to side n . If this output is greater than 0.5, a 1 is sent. Otherwise, a 0 is sent.
$y_n, n \in [6, 11]$	$[0,1]$	Attach/detach output for side $n - 6$. At each timestep, if a randomly generated value between 0 and 1 is less than the average of this output for two adjacent sides, their two respective molecules are attached. Otherwise, they are detached.
y_{12}	$[-1,1]$	Molecule rotation output. If $-0.33 < y_{12} < 0.33$, the molecule does not rotate. The remainder of the output range is equally divided to represent the four possible rotations, two directions per half.

 Table 1: Simulated molecule controller inputs x and outputs y .

neighbour's, the inherent bias in the cube evaluation order is lessened.

Experiments

The goal of the experiments presented in this paper was to observe self-replicating cube "species" in a simulated three-dimensional environment. Here, a replicating species is defined as a genome that occurs in two or more genetically homogeneous molecule conglomerations, where a molecule conglomeration is defined as a grouping of two or more attached molecules. Genetic distance was calculated as the sum of the tree edit distance between each output equation in a pair of genomes (tree edit distance was calculated using the Zhang-Shasha algorithm [Zhang and Shasha (1989)]). Self-replication is defined here as a series of actions whereby a genetically homogeneous molecule conglomeration accumulates molecules from the environment and/or other conglomerations, overwrites their genomes with its own and then detaches at one or more points to produce two or more genetically homogeneous conglomerations that all contain the same genome. Self-replicating species are detected by searching the simulation for genomes that exist in two or more distinct, genetically homogeneous conglomerations. Note that the structures of the molecule conglomerations are ignored in this definition. This is owing to the fact that while genetically identical conglomerations were often observed, they were usually composed of a different number of molecules, or the same number but arranged differently.

Experiments consisted of 1,000 randomly placed molecules, each with a randomly generated genome. Experiments were performed with densities of 0.25%, 1%, 4%, 16% and 64% (note that in cases other than 1% density, the number of cubes had to be adjusted slightly to achieve the desired density). The mutation rate used was $\mu = 0.01$. An experiment would run for 10,000 timesteps, where a timestep consists of evaluating every molecule's controller, executing their outputs and stochastically performing muta-

tions and equation overwrites. At periodic intervals, inter-conglomeration and between conglomeration genetic distances were calculated. If two or more genetically homogeneous conglomerations were found to contain the same genome, this species would be observed in a manually conducted test simulation. Test simulations would occur in smaller 3D grids (usually 9x9x9), populated by other conglomerations and/or single molecules extracted from the same original simulation. The test simulation would last for 1,000 timesteps, and the results would be visualized using an RGB colour scheme to represent relative genetic distances. The goal of these test simulations was to observe self-replication. Furthermore, a variety of quantitative metrics based on genetic distance were used to analyze the simulations and to detect and observe the emergence of self-replicating species.

Results and Discussion

For the following results, data were collected at 100 timestep intervals. Figure 2 shows, for all experiments, the number of different self-replicating species detected at a given timestep (top), the average size of replicating conglomerations (middle), as well as the maximum number of conglomerations belonging to a single replicating species (bottom).

At low densities, replicators must be mobile to acquire new molecules. At a density of 0.25%, very few replicating species arise, as there is little interaction between molecules. Replicating species do appear on occasion, however they cannot acquire new molecules fast enough to replicate further before succumbing to mutations. At 1% density, mobile conglomerations encounter new molecules more frequently. Initially, a few small replicators appear. Over time, these initial replicators collect stationary molecules, thus spreading genomes that promote conglomeration mobility. This also enables molecules that were initialized without immediate neighbours to interact with other cubes. Thus, the molecules in the system become more mobile, increas-

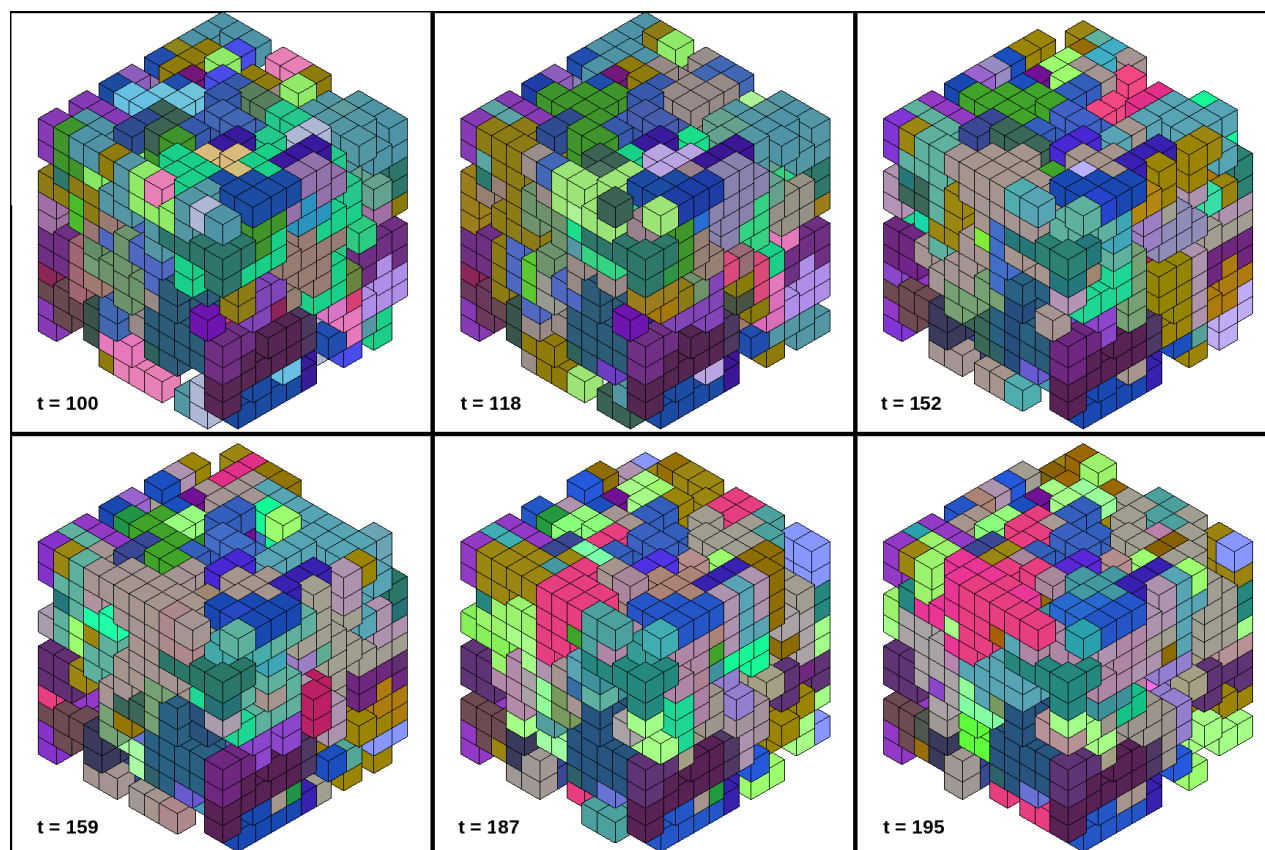


Figure 3: A collection of timesteps during a full simulation run at 64% density. Colours represent relative genetic distance. Despite their almost complete lack of mobility, several replicating species succeed in dominating large sections of the simulation world, albeit temporarily.

ing the number of molecule interactions, which in turn produces more replicating species and larger conglomerations.

At higher densities, molecules are more likely to be initialized with adjacent neighbours, therefore a large number of replicating species appear within the first 100 timesteps. Interestingly, the number of distinct replicating species decreases as the simulation progresses, with the higher density simulations (16% and 64%) finishing with less distinct species on average than the lower density simulations (1% and 4%). This is most likely owing to the larger number of molecule interactions that will occur at higher densities, which in turn will lead to more competition and a larger number of equation overwrites per timestep, thus reducing overall diversity. At a density of 64%, mobility is extremely limited. Regardless, self-replication emerges consistently, with larger species conglomerations on average. Figure 3 shows several timesteps of a 64% density simulation run.

Figure 4 compares the original results from the 1% density runs with a new set of results from a similar 1% density simulation where the only difference was that the outputs of all molecules were randomly generated values in

the range $[0, 1]$. These values were regenerated at each timestep. These data show that self-replicating species can occasionally arise from inherent properties of the simulation itself. However, these species are on average the minimum possible size (two molecules per conglomeration, the minimum number required to be defined as a conglomeration) and comprised of the minimum number of conglomerations (two conglomerations, the minimum number required to be defined as a species). Thus, while a minimal amount of self-replication can occur in the system by chance, having the genomes control the molecule outputs allows for a larger number of self-replicating species to emerge from the simulation. These genome controlled species are also on average more complex (i.e. more molecules per conglomeration) and more reproductively viable (i.e. produce more copies of themselves) than their randomly arising counterparts.

Figures 5 and 6 show two examples of test simulations where replication was observed. In both scenarios, the test grid was $9 \times 9 \times 9$ and all conglomerations were extracted from the same original 1% density simulation run. The conglom-

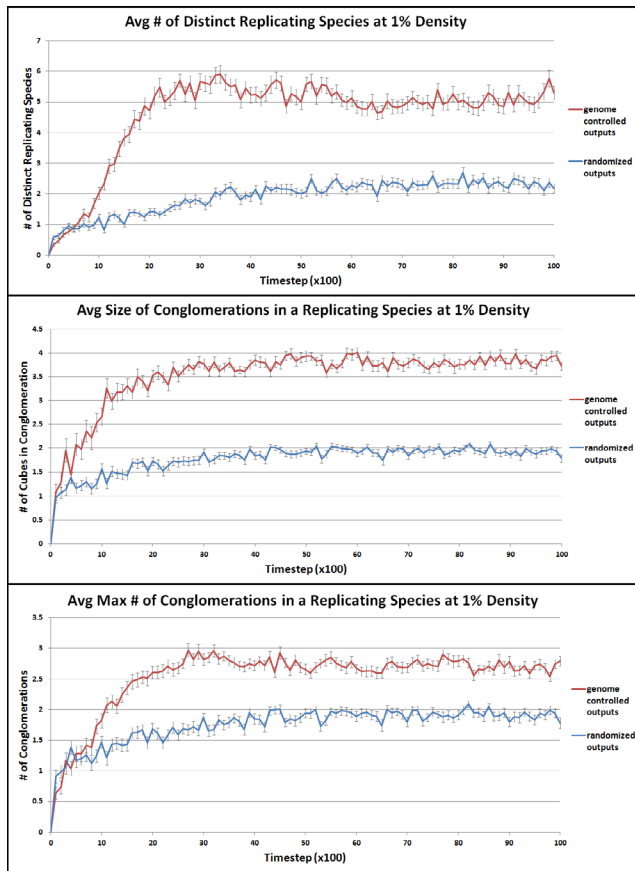


Figure 4: Experimental results for 1% density runs. “Genome controlled outputs” are the original results from the simulation as described. “Randomized outputs” are results from a simulation identical to the original, except that the outputs of each molecule were set to random values at each timestep. Each result is an average over 100 randomly initialized runs and error bars show standard error.

erations shown in Figure 5 were from timestep 7,900, while those in Figure 6 were taken from timestep 9,700. Figure 5 shows a large conglomeration dividing multiple times. It begins the test simulation composed of eight molecules, which was its structure when it was extracted from the original simulation. It splits almost immediately into two groups, one of three cubes and one of four, leaving a single cube unused. The conglomeration of size four soon splits again into two groups of two cubes. One of these two groups attaches to a genetically distinct conglomeration of size two and after a few timesteps of back-and-forth stochastic genetic exchange, it is able to overwrite the foreign genomes with its own, thus becoming a genetically homogeneous conglomeration of size four. By the end of the test run, the original conglomeration of eight cubes has replicated multiple times, with the help of two cubes consumed from a foreign conglomeration.

In Figure 6, the blue conglomeration with four cubes consumes the two cubes in the green conglomeration. It then moves on to attach itself to the orange conglomeration. Despite being one cube smaller, the stochastic overwrites work in the blue conglomeration’s favour, allowing it to rapidly overwrite the orange conglomeration. Finally, the single 13-cube conglomeration splits into two genetically identical conglomerations of size six, leaving a single cube unused. Thus in only 15 timesteps, the blue species was able to consume all of the molecules in the test simulation and use this material to self-replicate.

It seems counter-intuitive that self-replication would arise in so few timesteps considering the large number of inputs and outputs for a molecule controller. In low-density situations, a self-replicating conglomeration must be able to move through the simulated 3D world and attach to new molecules in ways that do not impede mobility. Moreover, at all densities, replicators must be able to detach at appropriate inter-cube connections and at appropriate times to produce viable copies. It turns out that a simple cube controller can produce these desired properties. For example, the controller in the blue cubes in Figure 6 is largely static, with the majority of its outputs set permanently to 0 or 1. This includes its turn output. Four of its six attach/detach outputs are static, with two set to 0 and two set to 1. The only fully dynamic outputs¹ are two of its attach/detach outputs, shown in simplified form in Equations 2 and 3.

$$dy_9/dt = dx_2/dt \quad (2)$$

$$dy_{11}/dt = \begin{cases} 0.074, & \text{if } x_7 = 0.0 \\ -0.77, & \text{if } x_7 = 0.5 \\ -0.54, & \text{if } x_7 = 1.0 \end{cases} \quad (3)$$

Thus, as in 2D cellular automata, a simple controller governing the interaction of multiple identical agents in a simulated 3D world can produce surprisingly complex behaviours. Note that the attach/detach output shown in Equation 2 depends on an incoming communication bit, demonstrating how communication bits can be used to decide when and where a cube conglomeration should split.

Conclusions and Future Work

As far as the authors know, the results presented in this paper are the first cases of the spontaneous emergence of self-replication in a simulated three-dimensional environment. Previous results (e.g. [Studer and Lipson (2005); Chou (1997)]) occurred in two-dimensional scenarios. Furthermore, by simulating molecules that have been constructed in the real world, we are one step closer to evolved,

¹“fully dynamic” outputs are ones that continue to change over time. This controller also had several “partially dynamic” equations that could change an output once before becoming static.

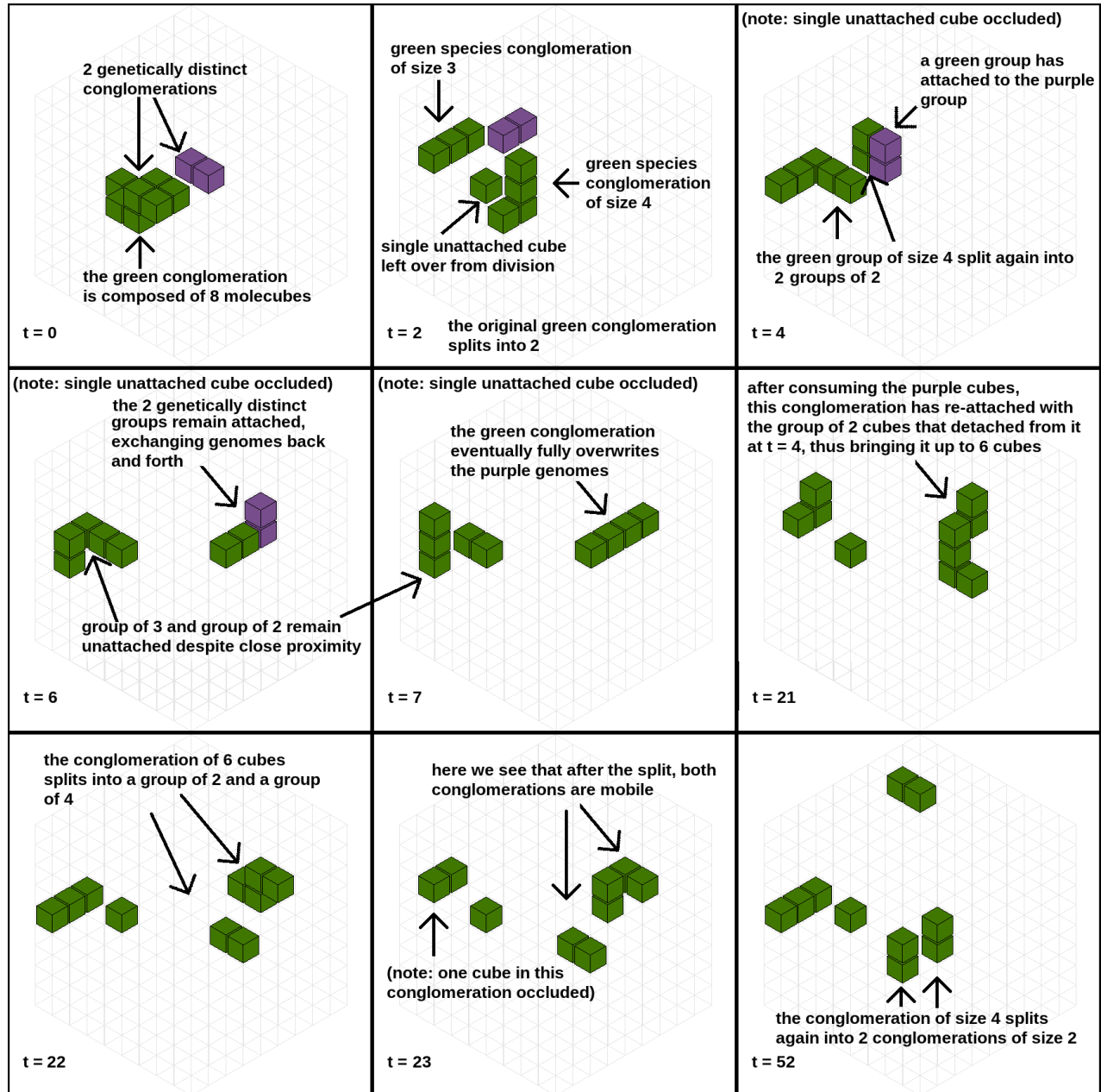


Figure 5: Test simulation using conglomerations from timestep 7,900. Colours represent relative genetic distance. A large conglomeration replicates multiple times. It also captures a small genetically distinct conglomeration and uses its cubes for self-replication.

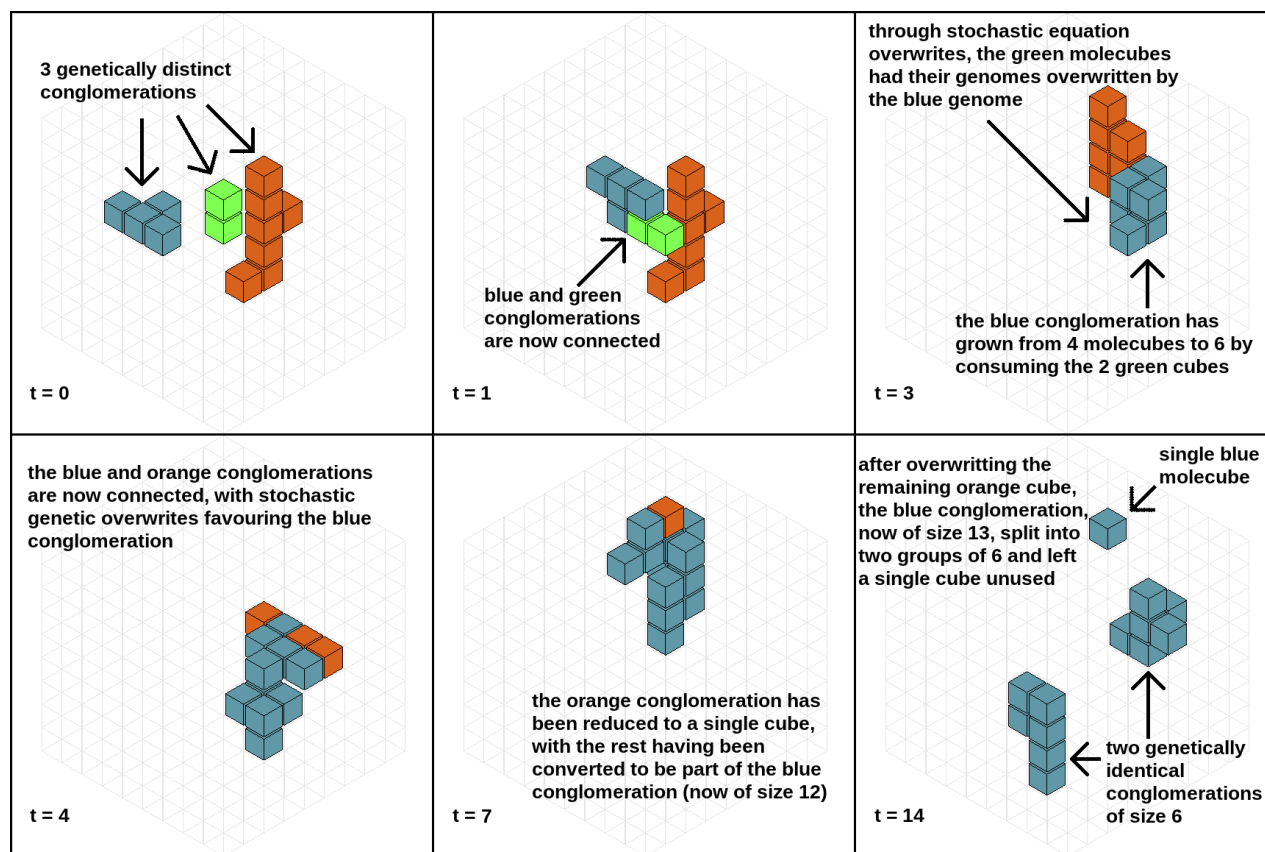


Figure 6: Test simulation using conglomerations from timestep 9,700. Colours represent relative genetic distance. The blue conglomeration consumes the other groups and uses their cubes to self-replicate.

physically realizable self-replicating machines. The next steps toward this goal would be to incorporate more physics into the simulation, including gravity, and to have the 3D simulated world be continuous instead of partitioned into a discrete grid.

In a 3D simulation, evolving controllers have a large number of inputs and outputs to contend with, and the number of potential situations in which a molecule conglomeration might find itself is very large. Future work should focus on further evolving these self-replicating species in an effort to produce species with more complex behaviours. Incorporating nature-inspired operations such as crossover and random death might help to increase the evolved capabilities of the controllers.

Despite the complexities associated with a three-dimensional world, a plethora of self-reproducing molecule conglomerations emerged in every run of our 3D simulation at densities of 1% and higher. Using simple, largely static controllers, these conglomerations were able to collect other molecules and use them to produce new, genetically identical conglomerations. The simplicity of the controllers coupled with the frequency of the emergence of self-replication

in scenarios requiring mobility as well as in scenarios that allowed for only limited mobility demonstrates that a diversity of surprisingly complex behaviours can emerge from the interactions of relatively simple agents in a simulated three-dimensional world.

Acknowledgements

This work was supported in part by the U.S. National Science Foundation's Office of Emerging Frontiers in Research and Innovation (grant number 0735953). Paul Grouchy would also like to thank the Natural Sciences and Engineering Research Council of Canada for its support through the Canada Graduate Scholarship and the Michael Smith Foreign Study Supplement.

References

- Adami, C. and Brown, T. C. (1994). Evolutionary learning in the 2d artificial life system *avida*. In Brooks, R. A. and Maes, P., editors, *Artificial Life IV*, pages 377–381, Cambridge, MA. MIT Press.
- Bonner, J. T. (1998). The origins of multicellularity. *Integrative Biology: Issues, News, and Reviews*, 1(1):27–36.

- Chou, H. (1997). Emergence of self-replicating structures in a cellular automata space. *Physica D: Nonlinear Phenomena*, 110(3-4):252–276.
- Grouchy, P. and D’Eleuterio, G. M. T. (2010). Supplanting neural networks with ODEs in evolutionary robotics. In *Proceedings of the 8th international conference on Simulated evolution and learning, SEAL’ 10*, pages 299–308, Berlin, Heidelberg. Springer-Verlag.
- Mytilinaios, E., Desnoyer, M., Marcus, D., and Lipson, H. (2004). Designed and evolved blueprints for physical self-replicating machines. In *in Proc. of the 9th Int. Conf. on the Simulation and Synthesis of Living Systems (Artificial Life IX)*. MIT, pages 15–20. MIT Press.
- Neumann, J. V. (1966). *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign, IL, USA.
- Poli, R., Langdon, W. B., and McPhee, N. F. (2008). *A Field Guide to Genetic Programming*. Lulu Enterprises, UK Ltd.
- Ray, T. S. (1992). An approach to the synthesis of life. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II*, pages 371–408. Addison Wesley Publishing Company.
- Studer, G. and Lipson, H. (2005). Spontaneous emergence of self-replicating, competing cube species in physical cube automata. In *Proceedings of the 2005 Genetic and Evolutionary Computation Conference, late breaking paper, Washington D.C.*
- Zhang, K. and Shasha, D. (1989). Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.*, 18(6):1245–1262.
- Zykov, V., Mytilinaios, E., Adams, B., and Lipson, H. (2005). Self-reproducing machines. *Nature*, 435(7039):163–164.

Open Ended Evolution of 3D Multicellular Development Controlled by Gene Regulatory Networks

Michał Joachimczak¹ and Borys Wróbel^{1,2,3}

¹Systems Modeling Laboratory, IO PAN, Sopot, Poland

²Evolutionary Systems Laboratory, Uniwersytet im. Adama Mickiewicza, Poznań, Poland

³Institut für Neuroinformatik, Universität & ETH Zürich, Switzerland

Abstract

We demonstrate how a novelty search algorithm can be used to create an open ended evolution system for 3-dimensional (3D) morphologies in which a constant evolutionary pressure exists for new shapes to be produced. In our platform, GREaNs, multicellular development starts from a single cell and all cells share the same genome and the same topology of the regulatory network. The size of the genome and the size of the network are not limited. Gene products can influence gene expression in the cells that produce them (such products act like transcription factors) or diffuse from one cell to another (acting like morphogens in biological development). We use the novelty search algorithm as a way to explore the space of achievable phenotypes in our platform for a given developmental setup. We analyze the evolutionary history in independent runs to see if a similar area of the phenotypic space is explored and discuss the features of evolutionary histories in the novelty search.

Introduction

Biological evolution is a process that resulted in a complex and intertwined history of all living organisms on our planet and current incredible diversity of life. This process inspired a commonly used optimization method, a genetic algorithm. The method relies on a formulation of a fitness function, which measures the quality of a particular solution (phenotype). However, biological evolution works differently. First of all, ancestors of currently living organisms had to compete with their contemporaries for limited resources in their environment, and the ability of these ancestors to reproduce in the environment in which their descendants live is irrelevant. But even disregarding the changing biotic and abiotic parts of any environment, at any particular time there are many ways in which a phenotype can affect the reproduction of genes which specify it, and the path along which optimization has occurred in a particular lineage may have been taken because of historical accidents. The changing environment and the fact that different aspects of the phenotype can be optimized contribute to the ability of evolving lineages of biological organisms to escape from the dead alleys (local optima in the fitness landscape).

Many optimization problems exhibit fitness landscapes in which genetic algorithms perform very poorly. This is especially the case when finding the optimal solution requires the search to proceed in a direction that is different than the local gradient of the fitness function. Such fitness landscapes are called deceptive. The novelty search algorithm, proposed by Lehman and Stanley (2011), is an evolutionary method that attempts to deal with this issue by avoiding the use of an explicit fitness function. Instead, the algorithm favors the individuals in the population that phenotypically differ the most from the other individuals in the current and past generations. Provided that the distance measure between individuals is relevant to the task at hand, the novelty search does not result in a blind walk through the search space. The method has been shown to outperform the evolutionary methods based on a fitness function in some problems (for example, evolving a control for a robot that moves through a labyrinth; Lehman and Stanley, 2011).

Novelty search differs from other approaches (see e.g., Mahfoud, 1995; Sareni and Krahenbuhl, 1998) to increase genetic diversity during evolutionary search, among which the fitness sharing is perhaps the most popular. The difference is that novelty search focuses entirely on the diversity of the phenotypes, not on the diversity of the genotypes. This requires a measure of distance between any two phenotypes. Lehman and Stanley (2011) define the measure of novelty as the average distance of an individual x to its k -nearest neighbors (a measure of sparseness of the phenotypic space surrounding the individual):

$$\rho(x) = \frac{1}{k} \sum_{i=0}^k d(x, \mu_i) \quad (1)$$

where μ_i is the i -th nearest individual to the one at hand according to the distance metric d .

The individuals used to compute the distance are recruited from the current population as well as from the past generations. The latter is important. Otherwise, the population could backtrack in the search space, rediscovering phenotypes that were found earlier. However, computing distance from all the past individuals would in many cases be

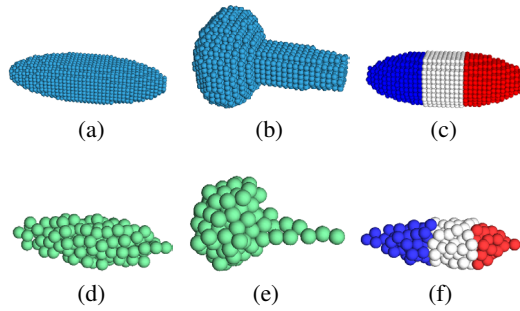


Figure 1: Examples of morphologies obtained using an objective fitness function and a genetic algorithm. (a,b,c) voxelized target shapes (small spheres represent voxels), (d,e,f) the best individuals in 10 independent evolutionary runs (spheres are cells).

computationally prohibitive. This is why only a selection of the past individuals is used. This selection is known as the archive. Whenever a considerably novel (with novelty above a threshold) phenotype is discovered, it is copied to the archive, and it remains there as a representative of its type.

In this paper we explore the possibility of using the novelty search in order to create an open-ended system for evolving 3D morphologies. We believe that in such a system evolutionary pressures are more similar to the pressures in biological evolution. The apparent complexity of morphologies that can be evolved using the novelty search quickly outreaches morphologies that we could obtain using a fitness based approach (Joachimczak and Wróbel, 2008, 2009; Fig. 1). This suggests that the novelty search can be a more appropriate way to explore what kind of morphologies are reachable in a given artificial embryogeny system than a genetic algorithm.

The Model

3D multicellular development controlled by a Gene Regulatory Network (GRN)

The network structure in our platform, GReaNs (which stands for *Genetic Regulatory evolving artificial Networks*) is specified by a linear genome without imposing any limit on the number of nodes and links or the size of the genome. The approach is similar to that used by Eggenberger Hotz (1997), and recently also by other authors (e.g., Schramm and Sendhoff, 2011) for modeling multicellular development. The network structure in all the cells is the same, but cells can differentiate because they may differ in the network state. The state of the network outputs determines if a cell divides or dies. The cells in GReaNs can move freely in a continuous 3D space, unlike in other systems of GRN-controlled development where a grid is used (e.g., Eggenberger Hotz, 1997; Kumar and Bentley, 2003; Cussat-Blanc

Algorithm 1: Decoding of the genome into the GRN.

1. For each series of 1+ P followed by 1+ G elements:
 - form a regulatory unit, a node in the GRN (N)
2. For each S element in the genome:
 - form an input node (I) or an output node (O) (depending on the order in the genome)
3. For each pair of nodes N_i, N_j :
 - consider the position of each P in N_i and each G in N_j , and if the distance is below a cut-off, make a link (L)
 - $\text{weight}(L)$ is an exponential function of the distance with maximum value of 10 for zero distance
 - the sign of $\text{weight}(L)$ is determined by the product of “sign” fields of both elements
4. For each input node I_i and each node N_j :
 - consider each P in N_j and make a link (or not) as in step 3
5. For each node N_i and each output node O_j :
 - consider each G in N_i and make a link (or not) as in step 3

et al., 2008; Chavoya et al., 2010). The evolvability in our system was investigated using a genetic algorithm to obtain artificial multicellular bodies with a specific 3D shape (Joachimczak and Wróbel, 2008) and pattern of gene expression (in the first successful attempt we are aware of at solving the so called “French flag problem” in 3D; Joachimczak and Wróbel, 2011; Fig. 1cf). We use in this work essentially the same model as we used before (Joachimczak and Wróbel, 2011), but we describe it here briefly for completeness.

A genome (Fig. 2) in GReaNs consists of regulatory units, each containing genetic elements, which come in several types, grouped into classes. One class of genetic elements (S) is reserved for elements that correspond to the GRN outputs or inputs, but the most important distinction is between class P (cis-regulators, which in biology are often close to promoters) and G that (like genes) encode trans-regulators. One type of trans-regulators can act only in the cell that produces them (they are like biological transcriptional factors), another can diffuse from one cell to another (like biological morphogens). At the beginning of the simulation, the genome is converted into a GRN (Algorithm 1).

In each simulation step, the concentration of each product is determined. All the products in the same unit have the same concentration. First, the promoter activation is calculated and converted into production/degradation rate with

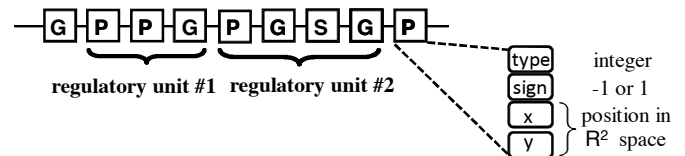


Figure 2: Genetic elements, regulatory units and the linear genome. Elements (left) have type (and class: S, P, G), sign, and coordinates (a position in 2D space).

Algorithm 2: Pseudocode for obtaining the concentration of products of each regulatory unit or node N at simulation step s , $\text{conc}(N, s)$.

```

foreach node  $N_i$  do
   $\text{activation} = 0$ ;
  foreach link  $L_k$  connecting  $N_j$  with  $N_i$  do
     $\text{activation} =$ 
       $\text{activation} + \text{weight}(L_k) \cdot \text{conc}(N_j, s - 1)$ ;
   $\text{rate} = \tanh(\text{activation}/2) - \text{conc}(N_i, s - 1)$ ;
  /*  $\text{dt}$  is the integration time step */
   $\text{conc}(N_i, s) = \text{conc}(N_i, s - 1) + \text{rate} \cdot \text{dt}$ ;

```

a sigmoid function. From the obtained value, the intrinsic degradation rate (equal in value to the current concentration) is subtracted. In other words, products degrade exponentially with time if the activation of the promoters is not high enough (Algorithm 2). S elements corresponding to inputs specify in effect products whose concentration is determined externally to the cell. On the other hand, when an output node is formed for an S element, the node acts as if it had one promoter (with the position corresponding to the position of the element) and one product.

The embryo growth starts from one cell (the zygote). If a specific product (specified by an output node) is above pre-set threshold in a particular cell (a mother cell), a new cell is formed (a daughter cell) and put close to the mother in the direction specified by the mother's "division vector". The product concentrations in the daughter are initially the same as in the mother, but the direction of the daughter's division vector may be modified at this point, depending on the concentration of three specific products (of output nodes). The daughter cell is pushed away from the mother by physical forces present in the environment. The physics includes repulsion when the cells are too close, adhesion between cells up to a certain distance, fluid drag to prevent erratic movements, and rules for a simplified model of diffusion. The model of diffusion ensures that a concentration of a particular morphogen in a given cell depends on the distance of this cell to the cell that produce the morphogen, with a delay in the propagation of the signal. In addition to morphogens produced by the cells, there are 5 additional diffusive substances present in the environment (coded by S elements): one has a uniform constant concentration, maximum allowed by the system (1), four others diffuse from specific points in space. External factors are, first of all, necessary to start the activity of the GRN in the zygote, secondly, they work in a similar fashion as a bias input in artificial neural network, and thirdly, the ones that are anisotropic help guide the cell differentiation.

Measure of distance between phenotypes

To calculate the novelty of each phenotype in every generation (Eq. 1), we used an approach based on the one used pre-

viously in a genetic algorithm to compare directly the phenotype and a target (Joachimczak and Wróbel, 2008, 2009). The distance between two individuals A, B :

$$d_{dir}(A, B) = \frac{1}{s_x s_y s_z} \sum_{x=0}^{s_x-1} \sum_{y=0}^{s_y-1} \sum_{z=0}^{s_z-1} |A_{xyz} - B_{xyz}| \quad (2)$$

is obtained by first discretizing each shape, then putting each in a cuboid with dimensions s_x, s_y, s_z , and finally calculating the number of different voxels (A_{xyz}, B_{xyz} is the voxel state at position x, y, z , 1 when filled, 0 when empty). The value of d_{dir} is usually small, because each shape occupies only a small fraction of the volume of the cuboid (which needs to be large to allow for a large spectrum of shapes).

The limitation of directly comparing the shapes in this way is that an absolute coordinate reference system is used. In effect, it is possible to obtain a large value of the distance for two individuals that appear visually similar, but whose development differs in the orientation of the division vector in some cells at the early stages of development. To avoid large distances for two shapes that differ by rotation, we perform a second comparison after putting each shape in the coordinate system defined by its principal components. In other words, Principal Component Analysis (PCA) is applied to a set of cell positions of each shape (A, B). This results in putting the shapes in reference systems with X axes aligned with the longest axes of the shapes. Then, the distance d_{dir} between the two rotated shapes (A_{rot}, B_{rot}) is calculated. Finally, we calculate which comparison (direct or after rotation) gives a smaller value:

$$d(A, B) = \min(d_{dir}(A, B), d_{dir}(A_{rot}, B_{rot})) \quad (3)$$

We compare the phenotypes both directly and after rotation because PCA-based approach may result in a large distance for two very similar shapes that become aligned along different directions. This is why the minimum difference between two comparisons was chosen as a distance between morphologies. We do not compare mirrored shapes (PCA does not give directions for principal axes) for the sake of simplicity, so it is possible for mirrored versions of similar morphologies to be included in the archive.

Novelty search for 3D shapes

The development was simulated for 400 time steps, but if there were any cell divisions after time step 300, an individual was removed from the population (the last 100 steps were set apart to allow the physics to move the cells to their final locations in the structure). If the zygote did not divide, the individual was also removed. If the embryo reached 100 cells, cell divisions were stopped.

The population size was kept constant at 300 individuals and evolution progressed through 5000 generations. The initial population was constructed by creating random genomes

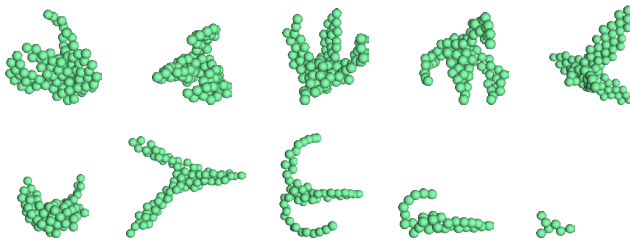


Figure 3: Morphological diversity in the final population. 10 individuals were selected from 300 in generation 5000 by visual inspection.

with a single regulatory unit, consisting of a single promoter and a single product. Although crossover was observed to improve evolvability in our previous work using a genetic algorithm (Joachimczak and Wróbel, 2009), it was disabled so that the full evolutionary history of any individual in the final generation could be traced backwards to a single ancestral individual in generation 0. Mutations could change the type, sign, and coordinates of genetic elements (changing affinities). Duplications (copying a group of elements and inserting it at a random position) and deletions were also allowed, with equal probabilities for both events.

The individuals were added to the archive either when they were novel or at random. The probability of random addition was set to $p = 5 \cdot 10^{-4}$. We have used a variable threshold (Lehman and Stanley, 2011) for the addition of random individuals (lowered when no additions during a certain number of generations, raised if too many).

Results and Discussion

A single evolutionary run using novelty search in GReaNs is enough to appreciate the morphological diversity which can be generated in the system. One way to have a glimpse at this diversity it to analyze the individuals in the final population (Fig. 3). Many of the structures have “appendages” and display radial symmetry. Our experience with simulating evolution of 3D morphogenesis using a genetic algorithm suggests that the complexity of many of these shapes is far beyond what is achievable using this previous approach (Fig. 1).

The collection of ancestors of the individual with the highest value of novelty in generation 5000 (Fig. 4) provides an example of an evolutionary trajectory. The evolution started from a spherical individual, but then “appendages” were added and modified over time. Individuals separated by a few hundreds of generations are still recognizable as variations of the same morphology or share some structural features. This indicates that there are no large random jumps in the exploration of phenotype space. Rather, evolution tends to progress through small phenotypic variations.

When the history of a whole run is analyzed, it can be seen

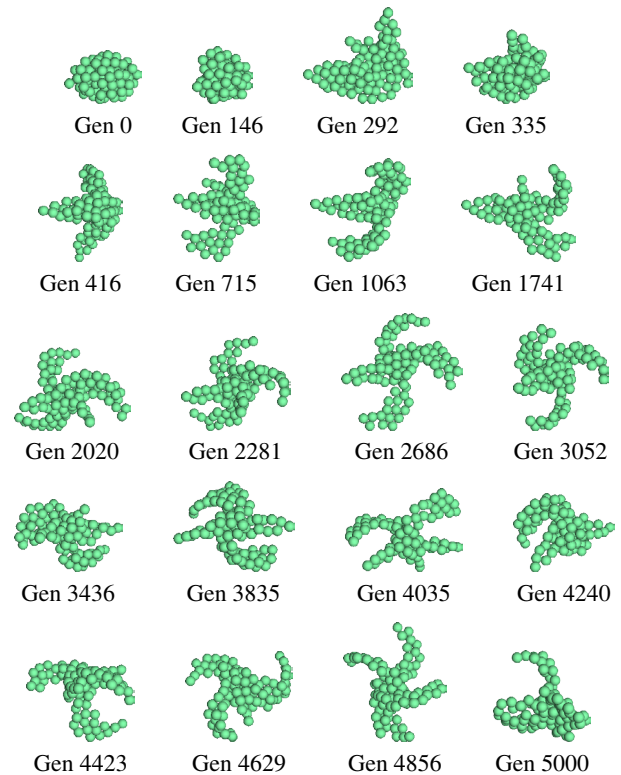


Figure 4: Selected direct ancestors of the individual with the highest value of novelty in generation 5000.

that the average level of novelty of the population (Fig. 5a) increased quickly in the first 500 generations, and then much more slowly during the remaining 4500. This does not indicate stagnation – if it happened, the novelty would decrease over time.

The analysis of genome size over time (Fig. 5b) suggests that the continuous generation of novelty stems at least in part from gene duplications. The fraction of non functional elements (TFs that do not bind to anything or promoters to which nothing can bind) remained relatively constant during the run, at the level of 15-30% (not shown), so the growth of the genome corresponded to the increase of the number of vertices in the regulatory network (Fig. 5c). The number of edges (Fig. 5d) stayed roughly proportional to the number of vertices. The initial values of the number of vertices are higher than 1, because apart from a single regulatory unit, the initial networks include a vertex for each input and output. The size of the genome and the network did not grow uniformly. For example, between generation 3700 and 4600 the average genome and network size dropped twofold, to later grow again. During this later growth the size of the network did not increase as much as the size of the genome, indicating accumulation of “junk” genetic elements.

The novelty search provides data that allows analysis of the entire evolutionary history, not only a single trajectory,

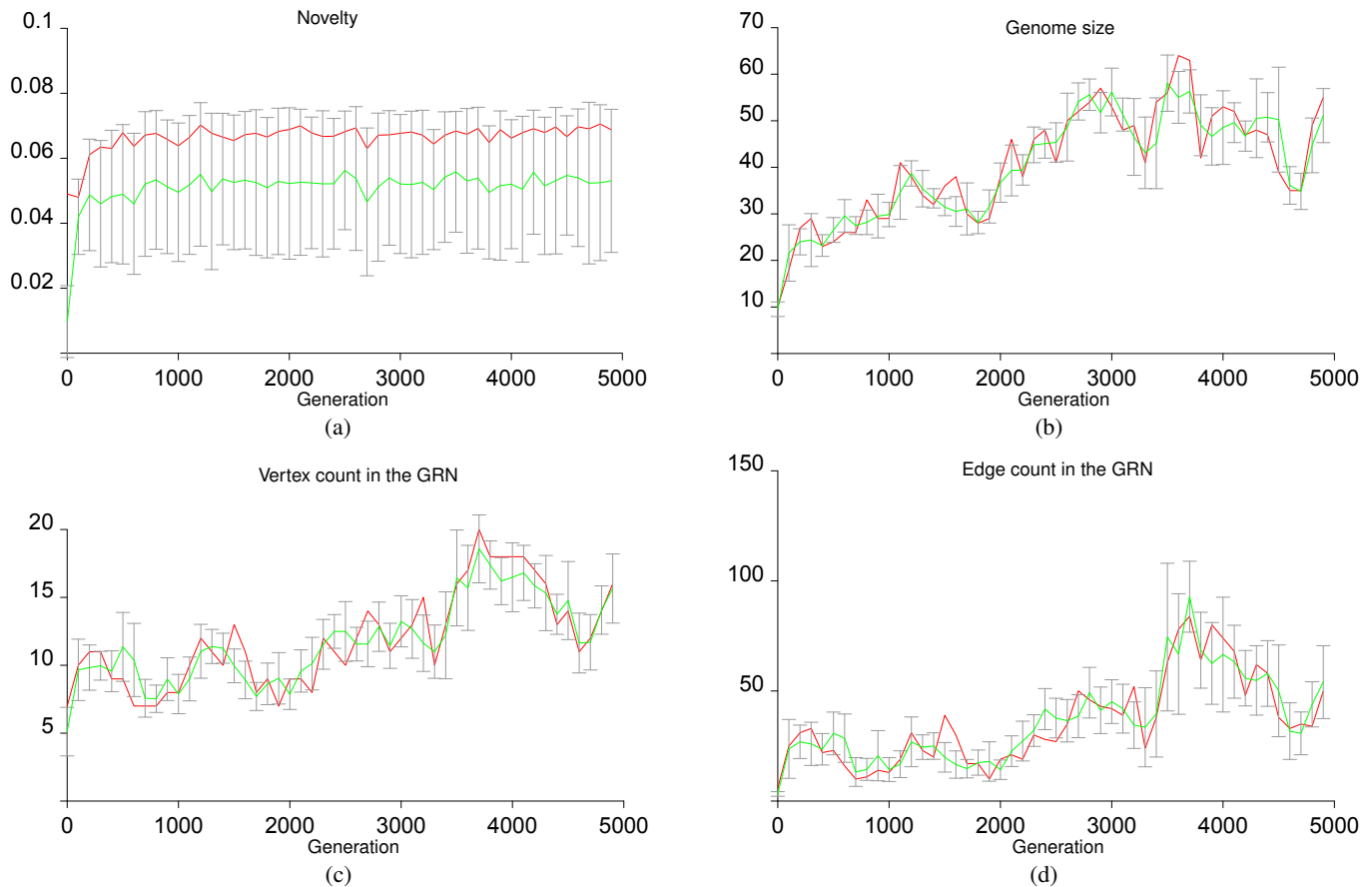


Figure 5: Novelty, the size of the genome and the regulatory network during the evolutionary run. Panel (a) shows novelty, (b) the number of genetic elements, (c) and (d) the number of vertices and edges in the gene regulatory network. The red line corresponds to the individual with the highest novelty in a given generation, the green line to the average (with bars indicating standard deviation). The values were determined every 100 generations.

because novel individuals are stored in the archive. The morphological diversity in the archive can be represented in a 2D space using multidimensional scaling (Fig. 6). We confirmed visually that neighboring points in such a representation correspond to similar morphologies. Strong patterns can be observed both for the individuals added to the archive because of their novelty (Fig. 6a) and randomly (Fig. 6b). First of all, in both cases many neighboring points come from generations that are not very far apart, which indicates that the jumps in the search space were not random. On the other hand, some individuals added later to the archive are similar to the ones added earlier. This can be explained by the fact that mutations in the genome specifying a more complex morphology can result in a simpler shape, similar to the phenotype of a genetically simpler ancestors. Such “degenerate” shapes include flat (the most salient clusters for both cohorts) and spherical phenotypes (which separate quite clearly from the shapes with “appendages” for random members of the archive; Fig. 6b). But, perhaps more interestingly, the other issue is that many complex forms (“body

plans”) appear to have emerged early on during the experiment. The fact that the novelty search revisits the corresponding areas in the phenotype space hints at a similarity between the evolutionary trajectories taken here *in silico* to what is thought to have happened during the evolution of life on Earth.

The generation-by-generation analysis of the cohort of the individuals added to the archive because of their novelty indicates that the evolution started with visually simple morphologies such as spherical clumps of cells or flat shapes (with a single layer of cells; Fig. 7). At that time, the genomes were still short (the initial random genomes had one single regulatory unit, basically allowing only for division). As the evolution progressed, more complex morphologies appeared, some with “appendages”. Some complex flat shapes were added to the archive, very likely created after a genetic element allowing to divide in the 3rd dimension was lost or damaged. Many morphologies look similar, possibly because once “appendages” appear, only small adjustments to genes controlling their growth are nec-

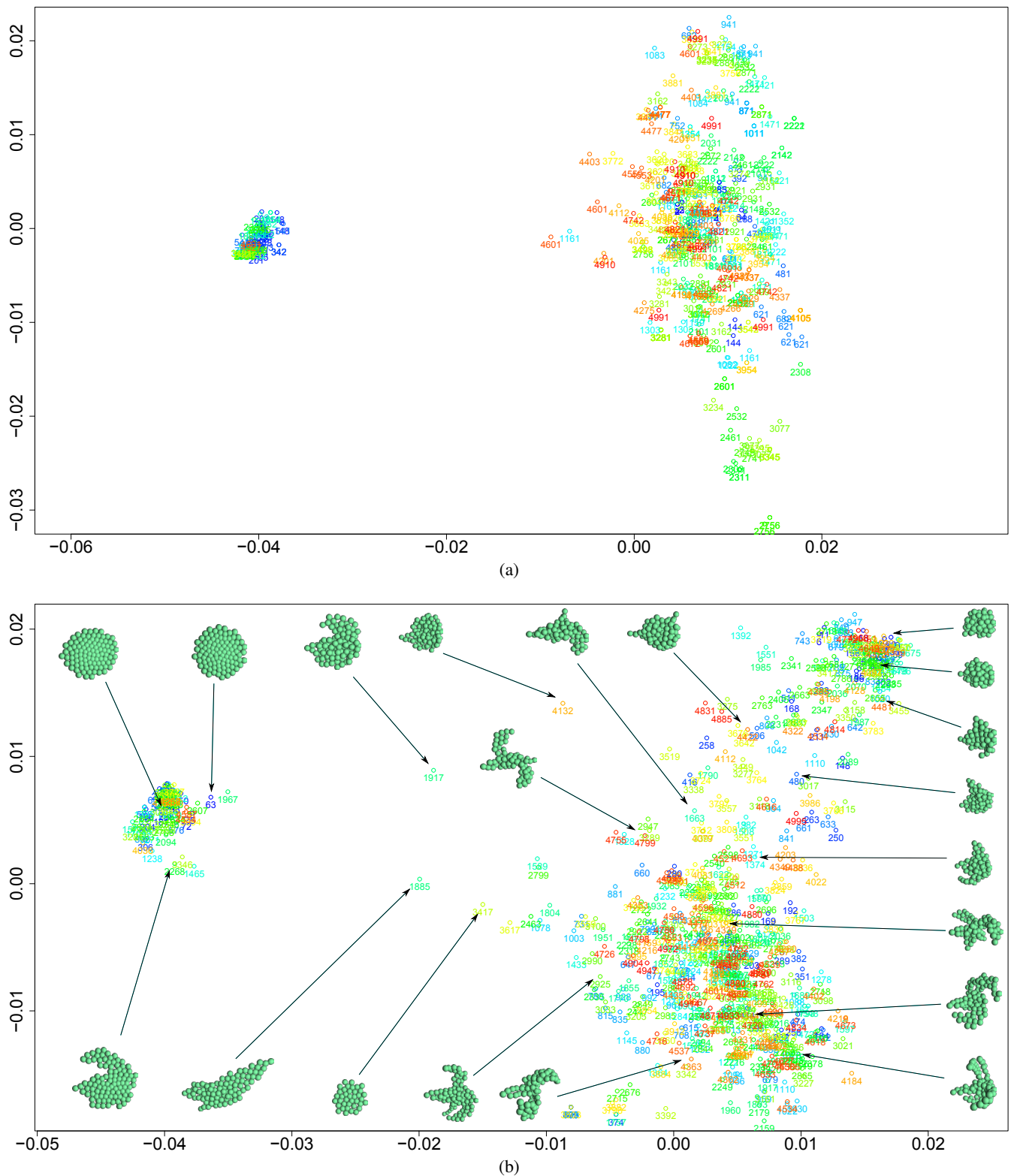


Figure 6: The phenotypic diversity of the individuals in the novelty search archive. Panel (a) shows multidimensional scaling of the distances (Eq. 3) between 432 individuals present in the archive at generation 5000, which were added to the archive because of their novelty, (b) shows the representation of 656 individuals added randomly. Each data point was labeled with generation number of the given individual and colored accordingly (blue: early individuals, green: intermediate, red: late).

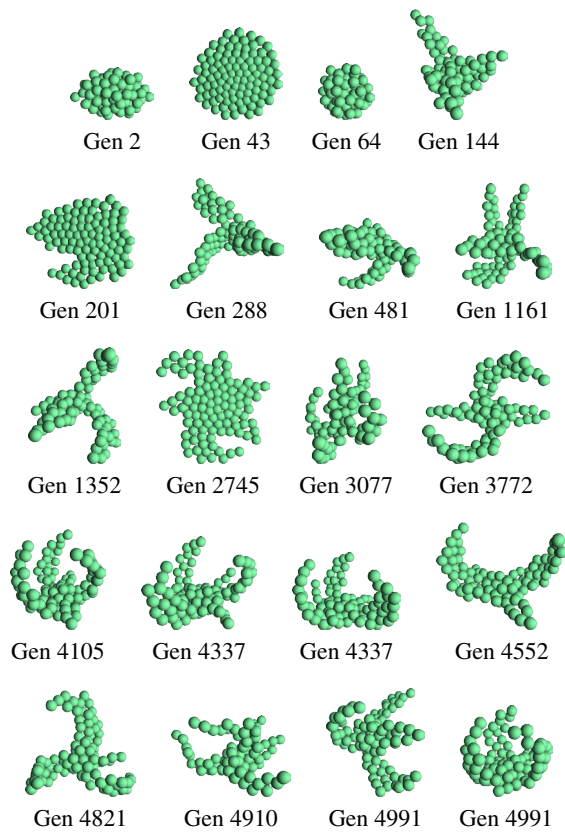


Figure 7: A sample of novel individuals in the archive at the final generation. The labels indicate the generation at which a given individual was added to the archive.

essary for a continuous stream of variation.

We have made several separate novelty search runs (starting each with a different seed for the pseudorandom number generator). Overall, the diversity of shapes obtained in such independent runs is similar (although individual evolutionary trajectories are, of course, different). Spherical or flat shapes appear initially, “appendages” later, with subsequent bending and twisting of “appendages” and changes in their number (Fig. 8). One mechanism for generation of novelty that was observed only in some runs is the development that employs cell death in the center of the embryo to arrive at disconnected morphologies (Fig. 8).

Conclusions and future work

Introduction of the novelty search in GReaNs allowed us to observe *in silico* an evolutionary process with features similar to biological evolution. We have observed some, but not all, of these features previously when a genetic algorithm was used in GReaNs, for example, the growth of the genome size over time. In the context of the novelty search, this growth suggests that gene duplications are important for the generation of morphological innovations. Similarly to the

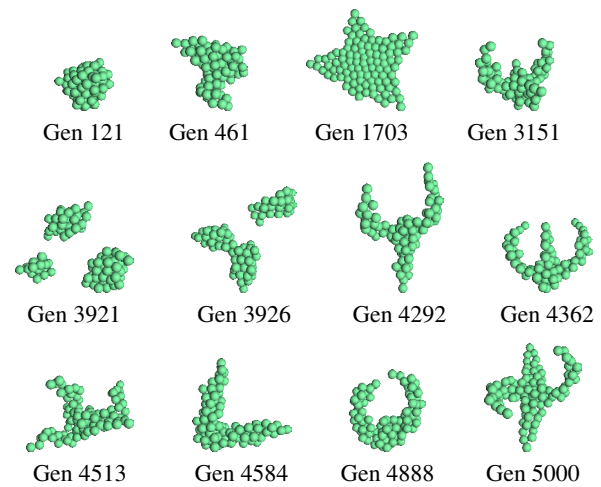


Figure 8: A sample of novel morphologies stored in the archive at the end of a separate evolutionary run. The labels indicate the generation at which a given individual was added to the archive. Around generation 3900, disconnected morphologies were added to the archive.

biological evolution, in the novelty search for 3D morphologies the phenotypic space is explored in small steps. Only over the long haul does the evolutionary trajectory of any individual contain ancestors which are very different from the forms in the later generations. Importantly, many of the morphologies obtained using the novelty search correspond to areas in the phenotypic space that are in practice, judging from our experience, unreachable by a target-driven genetic algorithm. The failure of the objective-driven search has been recently discussed in a system for evolving 2D patterns by Woolley and Stanley (2011), who compared a genetic algorithm with selective breeding (in which human guided the search selecting interesting shapes). We are planning to investigate this issue in more detail in GReaNs.

The search through the phenotype space with a novelty search algorithm produces emphatically different results than what would be expected from a random search. Random generation of genotypes in our system results mostly in individuals incapable of division or (at best) in very simple shapes: small clumps of cells, cells growing in a line. The novelty search avoids simple shapes like these because of the constant pressure to generate the morphologies which differ from what is currently in the population and what was seen in the past. The novelty search would also avoid completely “degenerate” shapes, for example, non-dividing individuals (consisting of one cell). In the experiments described here, the individuals which lost the product allowing for division were always removed from the population to speed up the evolutionary process. One could, in principle, devise some other criteria which might affect individual viability (for ex-

ample, the connectedness of the multicellular structure, or structural stability) in order to obtain shapes with desired functionality.

The design of the distance measure for phenotypes is at the core of the novelty search algorithm, much like the fitness function lies at the core of the genetic algorithm. The voxel-based approach used in this work is sensitive to minor changes in the angles of “appendages”. It would be interesting to test other measures of similarity of 3D shapes which would be less sensitive to that, and hence would put more pressure on generating new body plans. We could also formulate a measure that would force the search to avoid certain regions of the search space (for example, all flat individuals could be considered to be close), or to go in a certain direction (for example, promoting asymmetry or structures that are able to support themselves under gravitation).

The results presented here show how an existing developmental system can be modified to use a novelty search algorithm (by redefining the fitness function), leading to a creation of an open ended evolution system for 3D morphologies, in which constant evolutionary pressure exists for new shapes to be produced. Morphologies evolved in GReaNs, both with a genetic algorithm (Joachimczak and Wróbel, 2008, 2009) and with the novelty search, have comparable or indeed larger complexity than the morphologies evolved by other authors using GRN-based systems (cf. e.g., Eggenberger Hotz, 1997; Bongard and Pfeifer, 2001; Kumar and Bentley, 2003; Cussat-Blanc et al., 2008; Chavoya et al., 2010), although some developmental systems not based on GRNs allow for much higher complexity (e.g., Fontana and Wrobel, 2011). Our results indicate that incorporation of the novelty search algorithm in a developmental system can be seen as a way to explore the space of achievable phenotypes. It can also be seen as a way to transform the system into a more biologically plausible model. We believe that both issues are relevant for any model for the evolution of development and for further investigations on the relationship between evolution of the genomes, regulatory networks and morphological features.

Acknowledgments

Computational resources were provided by the Polish Ministry of Science and Education (project N519 384236, N303 291234), the Tri-City Academic Computer Center (TASK), and the Interdisciplinary Center for Molecular and Mathematical Modeling (ICM, University of Warsaw; project G33-8).

References

- Bongard, J. C. and Pfeifer, R. (2001). Repeated structure and dissociation of genotypic and phenotypic complexity in artificial ontogeny. In *GECCO 2001: Proceedings of The Genetic and Evolutionary Computation Conference*, pages 829–836. Morgan Kaufmann publishers.
- Chavoya, A., Andalon-Garcia, I. R., Lopez-Martin, C., and Meda-Campaña, M. E. (2010). Use of evolved artificial regulatory networks to simulate 3D cell differentiation. *BioSystems*, 102(1):41–48.
- Cussat-Blanc, S., Luga, H., and Duthen, Y. (2008). From single cell to simple creature morphology and metabolism. In *Artificial Life XI: Proceedings of the 11th International Conference on the Simulation and Synthesis of Living Systems*, pages 134–141. MIT Press, Cambridge, MA.
- Eggenberger Hotz, P. (1997). Evolving morphologies of simulated 3D organisms based on differential gene expression. In *Proceedings of the 4th European Conference on Artificial Life*, pages 205–213. MIT Press, Cambridge, MA.
- Fontana, A. and Wrobel, B. (2011). Epigenetic tracking: an evolutionary-developmental approach to generate very large complex systems. In *Proceedings of DevLeaNN: A Workshop on Development and Learning in Artificial Neural Networks*, pages 29–31.
- Joachimczak, M. and Wróbel, B. (2008). Evo-devo *in silico*: a model of a gene network regulating multicellular development in 3D space with artificial physics. In *Artificial Life XI: Proceedings of the 11th International Conference on the Simulation and Synthesis of Living Systems*, pages 297–304. MIT Press, Cambridge, MA.
- Joachimczak, M. and Wróbel, B. (2009). Complexity of the search space in a model of artificial evolution of gene regulatory networks controlling 3D multicellular morphogenesis. *Advances in Complex Systems*, 12(3):347–369.
- Joachimczak, M. and Wróbel, B. (2011). Evolution of the morphology and patterning of artificial embryos: scaling the tricolour problem to the third dimension. In *ECAL 2009: Proceedings of the 10th European Conference on Artificial Life*, volume 5777 of *Lecture Notes in Computer Science*, pages 33–41. Springer.
- Kumar, S. and Bentley, P. (2003). Biologically inspired evolutionary development. In Tyrrell, A., Haddow, P., and Torresen, J., editors, *Evolvable Systems: From Biology to Hardware*, volume 2606 of *Lecture Notes in Computer Science*, chapter 6, pages 57–68. Springer Berlin / Heidelberg, Berlin, Heidelberg.
- Lehman, J. and Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223.
- Mahfoud, S. (1995). *Niching methods for genetic algorithms*. University of Illinois at Urbana Champaign.
- Sareni, B. and Krahenbuhl, L. (1998). Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 2(3):97–106.
- Schramm, L. and Sendhoff, B. (2011). An animat’s cell doctrine. In *ECAL 2011: Proceedings of the 11th European Conference on the Synthesis and Simulation of Living Systems*, pages 739–746. MIT Press, Cambridge, MA.
- Woolley, B. G. and Stanley, K. O. (2011). On the deleterious effects of a priori objectives on evolution and representation. In *GECCO 2011: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, pages 957–964. New York, NY, USA. ACM.

Beyond Open-endedness: Quantifying Impressiveness

Joel Lehman and Kenneth O. Stanley

University of Central Florida, Orlando, FL 32826
jlehman@eecs.ucf.edu, kstanley@eecs.ucf.edu

Abstract

This paper seeks to illuminate and quantify a feature of natural evolution that correlates to our sense of its intuitive greatness: Natural evolution evolves *impressive* artifacts. Within artificial life, abstractions aiming to capture what makes natural evolution so powerful often focus on the idea of *open-endedness*, which relates to boundless diversity, complexity, or adaptation. However, creative systems that have passed tests of open-endedness raise the possibility that open-endedness does not always correlate to impressiveness in artificial life simulations. In other words, while natural evolution is both open-ended and demonstrates a drive towards evolving impressive artifacts, it may be a mistake to assume the two properties are always linked. Thus to begin to investigate impressiveness independently in artificial systems, a novel definition is proposed: Impressive artifacts readily exhibit significant design *effort*. That is, the difficulty of creating them is easy to recognize. Two heuristics, rarity and re-creation effort, are derived from this definition and applied to the products of an open-ended image evolution system. An important result is that the heuristics intuitively separate different reward schemes and provide evidence for why each evolved picture is or is not impressive. The conclusion is that impressiveness may help to distinguish open-ended systems and their products, and potentially untangles an aspect of natural evolution's mystique that is masked by its co-occurrence with open-endedness.

Introduction

A significant challenge in artificial life is to create an evolutionary system with dynamics and products similar in spirit to those of natural evolution. Some researchers believe that a truly *open-ended* evolutionary system will be a critical step towards that goal (Bedau et al., 1998; Standish, 2003). Though the definition of such open-endedness is still debated (Bedau et al., 1998; Lehman and Stanley, 2011a; Maley, 1999; Standish, 2003), there are a variety of reasonable intuitions about what constitutes open-endedness, e.g. increasing complexity, diversity, accumulation of novelty, and continual adaptation. Such intuitions typically are inferred from widely-accepted examples of open-ended evolution like natural evolution or the evolution of technology.

Some have attempted to quantify these intuitions (Bedau et al., 1998; Standish, 2003). Evolutionary activity statistics (Bedau et al., 1998) are the most popular of such measures,

and have been applied to many artificial life simulations (Bedau et al., 1997, 1998; Channon, 2001; Maley, 1999; Taylor and Hallam, 1998). The main idea motivating activity statistics is that an unboundedly open-ended evolutionary system will continually accumulate and preserve new adaptations. However, while several systems have passed the test (Channon, 2001; Maley, 1999), they do not seem to meet the high standard set by evolution in nature. The problem is that while the test indicates that adaptations accumulate, it does not reveal their *purpose*. As a result, it is difficult to decide whether the products of such systems are increasingly *impressive* (Channon and Damper, 2000; Maley, 1999). In other words, an increasing diversity of adaptations may not be a sufficient condition for what we appreciate intuitively about natural evolution. This possibility hints that open-endedness and impressiveness may not always be linked.

Approaching intuitions about evolution from a different perspective, this paper argues that a key feature of impressive open-ended systems like natural evolution is that their *products* are indeed impressive. For example, consider the human brain or the wide variety of complex animals crafted by natural evolution. Among their many features, they are usually regarded as impressive achievements. Yet what does impressiveness actually mean? Well-adapted natural organisms, elegant technological innovations, masterful human paintings, and great musical compositions all share the property that they are *easier to appreciate than to create*. Similarly to the concept of NP-completeness, wherein a computational solution is easy to verify but difficult to derive, this paper posits that impressive artifacts are those that readily exhibit significant design effort. In other words, it is easy to appreciate for an impressive creation how difficult recreating an artifact with similar properties would be.

This new formalization leads to two heuristics for quantifying the impressiveness of evolved products, *rarity* and *re-creation effort*, which are applied in this paper to an exploration-driven picture-evolution system. The results establish that the system discovers increasingly impressive artifacts compared to a random search or a direct search for rare artifacts. Importantly, what in particular makes

an evolved picture impressive is inherent in the introduced heuristics. In this way, the judgment of individual products of an artificial life simulation can be justified without appealing to subjective description. The main conclusion is that impressiveness illuminates a quantifiable facet of creative systems perhaps independent of open-endedness, one that may more deeply connect with what fascinates us about natural evolution.

Background

Because this paper introduces impressiveness, which is a measure related to open-ended evolution, this section reviews previous efforts to quantify open-ended evolution and prior investigations of concepts related to impressiveness. Novelty search, which is an approach to open-ended evolution applied in this paper's experiment, is also discussed.

Quantifying Open-Ended Evolution

In accordance with the general drive in science to formalize intuitions, there have been several attempts to quantify open-endedness (Bedau et al., 1998; Nehaniv, 2000; Standish, 2003). Such formalizations derive from intuitive features of open-ended systems, such as their drive towards diversity or complexity (Nehaniv, 2000; Standish, 2003), or their accumulation of adaptations (Bedau et al., 1998).

The dominant approach to quantifying open-ended evolution in artificial life systems is a particular measure of adaptation called *evolutionary activity statistics* (Bedau et al., 1998). The idea is that continual adaptation is a critical facet of open-ended evolution, and that persistence of traits in the face of selection is a proxy for measuring adaptation.

However, an interesting question is whether passing the activity statistics test is sufficient to equate an artificial system's creativity with that of natural evolution. Indeed, some systems have passed the test (Channon, 2001; Maley, 1999). Yet Maley (1999) acknowledges that his proposed systems will never create anything surprising and fall far short of intuitions about nature. Similarly, Channon and Damper (2000) note that in their system it eventually becomes difficult to describe what distinguishes new adaptations. In other words, passing the activity statistics test may establish open-endedness but it does not unambiguously demonstrate that a system continues to create interesting or impressive artifacts. Thus to facilitate investigating both the impressiveness of individual evolved artifacts and the tendency of artificial life simulations to create increasing impressiveness, this paper formalizes and suggests heuristics for impressiveness.

Impressiveness and Interestingness

The concept of impressiveness described in this paper also relates to the concepts of interestingness and beauty; intuitively, interesting or beautiful artifacts often tend to be impressive as well. Because they are general and important concepts, beauty and interestingness have previously been explored in diverse contexts including philosophy (Neill and

Ridley, 1995), reinforcement learning (Schmidhuber, 2009), and even data mining (Geng and Hamilton, 2006).

Though they overlap in some ways, a key difference between interestingness and impressiveness is that interestingness is often tied to time-dependence or novelty (Geng and Hamilton, 2006). That is, an object that is initially found interesting may become less interesting over time due to habituation. In contrast, the formalization of impressiveness in this paper is not relative to what has been observed before. For example, the human brain will always be an impressive artifact, although by some definitions of interestingness it becomes increasingly less interesting after repeated exposure. While the term *interesting* may also sometimes be applied in a time-independent context, the term *impressiveness* explicitly disambiguates the two usages and alleviates any confusion from overlapping colloquial usage. The important point is that because the notion of impressiveness expressed here is not a relative measure it can objectively compare results *between* experiments and not only *within* them.

In addition to relating to interestingness, impressiveness might also be seen as relating in some way to beauty; for example, Schmidhuber (2009) suggests both beauty and interestingness are rooted in compressibility. The idea is that the most compressible version of an artifact may be the most beautiful. In contrast, this paper relates the concept of impressiveness to the asymmetry between ease of recognition and difficulty in creating artifacts. Importantly, it is possible that what is most impressive or beautiful about an artifact may be mostly orthogonal to compressing it; for example, aesthetic qualities such as soft, vibrant, or ornate may summarize important facets of what is appreciated about a painting without reflecting how to reconstitute it from such properties. That is, compression is typically reversible to some degree while impressive properties may be approximately one-way transformations: easy to observe but hard to create.

The next section reviews novelty search, an algorithm designed for open-ended exploration that is applied to evolving pictures in the experiment in this paper.

Novelty Search

In contrast to most EAs, which tend to converge, novelty search is a *divergent* evolutionary technique. It is inspired by natural evolution's drive to novelty, and directly rewards novel behavior *instead* of progress towards a fixed objective (Lehman and Stanley, 2008, 2011a). Thus it matches well with artificial life domains that are not motivated by a defined set of objectives. This paper will ask whether the products of novelty search are impressive.

Tracking novelty requires little change to any evolutionary algorithm aside from replacing the fitness function with a *novelty metric*, which measures how different an individual is from other individuals, thereby creating a constant pressure to do something new. The key idea is that instead of rewarding performance on an objective, novelty search re-

wards diverging from prior behaviors. Therefore, novelty needs to be *measured*.

The novelty metric characterizes how far away the new individual is from the rest of the population and its predecessors in *behavior space*, i.e. the space of unique behaviors. A good metric should thus compute the *sparseness* at any point in the behavior space. Areas with denser clusters of visited points are less novel and therefore rewarded less.

A simple measure of sparseness at a point is the average distance to the k -nearest neighbors of that point. Intuitively, if the average distance to a given point's nearest neighbors is large then it is in a sparse area; it is in a dense region if the average distance is small. The sparseness ρ at point x is given by

$$\rho(x) = \frac{1}{k} \sum_{i=0}^k \text{dist}(x, \mu_i), \quad (1)$$

where μ_i is the i th-nearest neighbor of x with respect to the distance metric *dist*, which is a domain-dependent measure of behavioral difference between two individuals in the search space. Candidates from more sparse regions of the behavior space then receive higher novelty scores.

If novelty is sufficiently high at the location of a new individual, i.e. above some minimal threshold ρ_{min} , then the individual is entered into the permanent archive that characterizes the distribution of prior solutions in behavior space. The current generation plus the archive give a comprehensive sample of where the search has been and where it currently is; that way, by attempting to maximize the novelty metric, the gradient of search is simply towards what is *new*, with no other explicit objective.

Once objective-based fitness is replaced with novelty, the underlying evolutionary algorithm operates as normal, selecting the most novel individuals to reproduce. Over generations, the population spreads out across the space of possible behaviors.

Instead of rewarding novel agent behaviors as in prior novelty search experiments, in this paper novelty search explores a space of *image properties*, which can be conceived as the behaviors of neural networks asked to draw pictures. In effect this approach rewards novel pictures that exhibit characteristics different from those previously encountered.

Defining Impressiveness

It is often said that the artifacts evolved by natural evolution are impressive, as are many human innovations (Darwin, 1859; Kelly, 2010). In fact, such impressiveness may be intimately connected to our appreciation of such open-ended systems. However, it is sometimes unclear whether the products of artificial systems are similarly impressive. For example, some systems have passed the evolutionary activity statistics tests designed to validate open-ended evolution (Channon, 2001; Maley, 1999) yet few researchers have accordingly concluded that recreating the dynamics of natural

evolution is a solved problem. Such a discrepancy suggests that while activity statistics can successfully detect adaptation and perhaps an aspect of open-endedness, the mystery of prolific creative systems may run deeper than adaptation or open-endedness alone. In particular, an impressive open-ended system should also produce impressive artifacts. Thus a measure of impressiveness may serve as a new tool to help investigate open-ended systems.

Importantly, creating such a measure requires a definition that captures intuitions about what impressiveness means. The insight in this paper is that impressive artifacts exhibit significant design effort and that it is *easy* to recognize how difficult they were to create. To illustrate this idea, consider a gymnast performing a backflip in front of an observer.

Most observers would conclude the backflip was impressive because it takes significant strength and dexterity to defy gravity while completing a full airborne rotation and still landing squarely without falling. The general mechanisms underlying such judgments can be separated into two interrelated issues, first of mapping an observed event or artifact into an abstract description and then of judging how impressive that abstract description is. For example, the observer first recognizes the action of the gymnast as a backflip, and then evaluates how impressive a backflip is.

More specifically, the backflip is first recognized by the observer's visual system. Importantly, all that matters in observing that a backflip has occurred is that the gymnast jumps and completes a full rotation backwards in the air before successfully landing. In other words, the observer has extracted from a complex stream of sensory information a concise description that may be potentially impressive.

Once recognized, the complementary task is to judge the difficulty of this abstract description of a backflip. That is, an observer's internal understanding of physics and the athletic capabilities of most humans allows them to conclude reasonably that performing a backflip is challenging.

These two aspects combine to allow the observer to *recognize* how much effort is required to perform the action. Notice the fundamental asymmetry between recognizing and performing: It is much easier to appreciate a beautiful novel or a masterpiece than it is to create one. Interestingly, impressiveness is not a relative measure in principle. Even though it now requires less effort to create a machine that flies than it did in antiquity, the cumulative string of ideas that led to understanding flight will always be part of the true calculation of mechanized flight's impressiveness. However, in practice impressiveness may only be tractable when considered relative to a particular context (e.g. flight is not as impressive as it once was given an understanding of modern physics) or to a particular heuristic used to estimate it (e.g. re-creation effort, which is introduced later); similar practical limitations exist for other measures (Bedau et al., 1998).

Importantly, as it relates to artificial life, an impressive evolved artifact or organism will have recognizable proper-

ties that are difficult to recreate from scratch. For example, the functionality of a virtual creature might be impressive; it might locomote bipedally at a high speed, which would take many generations of evolution to achieve again. Notably, verifying an organism's speed is much simpler than creating an organism that travels at a high speed. In this way the concept of impressiveness relates to that of NP-completeness: Verifying solutions to NP-complete problems requires only polynomial computation while most researchers assume computing the solutions is impossible in polynomial time (Gasarch, 2002). Thus impressiveness can be defined as the difficulty of recreating an easily-recognized property of an artifact.

Measuring Impressiveness

The approach to investigating open-ended evolution in this paper is to measure the impressiveness of evolved artifacts. Thus this section introduces two heuristics derived from the definition of impressiveness proposed in the prior section. While it may be intractable in general to measure exactly how difficult a given property is to recreate, there are intuitive heuristics that may often reflect difficulty in practice.

The first simple such heuristic is *rarity*. That is, a property that can only be found in very small pockets of a large space may also be difficult to achieve. For example, few people are able to do backflips, which suggests it may be impressive. Similarly, few paintings are masterpieces and few novels are timeless. However, this heuristic is not without flaws because not all rare properties are hard to achieve. For instance, a person may have an odd quirk that no one else cares to acquire; though it is rare, acquiring that quirk may prove easy if attempted. Thus it is not really impressive. A more concrete example of this phenomenon can be given in the context of evolutionary algorithms. Imagine the space of all 100-digit binary numbers. Although the number consisting of all 1's is rare (occurring only once in 2^{100} possibilities), optimizing for such a property with a standard genetic algorithm is relatively trivial (Reeves, 2000). The fitness function of 1's in a given bit-string is not deceptive and is easily maximized.

Interestingly, this idea of optimizing for a particular property suggests a second, more rigorous heuristic: *re-creation effort*. If a property can be measured on a continuum, then the impressiveness of a particular level of that property can be estimated by applying a benchmark optimization algorithm to re-create that level. In other words, the difficulty for the benchmark optimizer to re-create an observed property of an evolved artifact is another way of estimating its impressiveness. Of course, the benchmark algorithm that defines the level of effort must be chosen carefully to obtain a reasonable *estimate* of the effort needed to discover a particular artifact. For example, evolving a virtual creature to reach a particular speed through a reasonable optimization algorithm may require on average a significant amount

of evaluations; therefore such quick locomotion may be impressive. Relating this heuristic to the backflip example, the amount of training required for the average person to learn how to do a backflip is significant.

Both of these heuristics are applied to investigate the products of the open-ended picture evolution system that is described in the next section.

Picture Evolution Experiment

An appropriate test domain for measuring impressiveness should potentiate both open-ended discovery and achieving impressiveness. Furthermore, there can be ambiguity within the results as to whether anything of interest has really occurred. In this way, the test domain may reflect a typical artificial life system wherein interpreting its products often appeals to subjective description. The motivation is that impressiveness can instead ground such results objectively through revealing *why* particular products are impressive.

A simple such domain is evolving pictures. The phenotype space of possible pictures is vast: A square image induces c^{n^2} possibilities, where c is the number of shade gradations for a single pixel and n is the size in pixels of one dimension. Also, humans intuitively appreciate many different properties of such pictures, e.g. their dominant color, level of symmetry, or smoothness. Furthermore, some combinations of such properties may be difficult to craft, especially when they conflict. For example, a picture with a low level of smoothness that still maximizes symmetry may require some aesthetic and technical skill to draw and thus may be more impressive than other pictures.

However, because aesthetic preferences for pictures are subjective and largely variable, judging the success of a given picture evolution system may be particularly contentious. That is, people may prefer different properties of pictures, which may cause them to disagree over whether a picture-evolving system has been successful or produced anything meaningful. However, a measure of impressiveness may be able to ground statements made about evolved pictures by indicating the degree of impressiveness and *what* about particular pictures is impressive.

Following the definition of impressiveness, to fit the measures of impressiveness to picture evolution it is necessary to identify potentially impressive properties of pictures that are easily recognizable. While humans are naturally able to recognize a wide range of picture attributes, such as symmetries, similarity to real-world objects, and various aesthetic qualities, a smaller set of properties is chosen for this experiment. The motivation is to create a reasonably-sized abstract space of picture characteristics that would serve both as a basis for recognizing impressiveness and as a behavior space for novelty search to explore.

Note that although the term *space* most frequently refers to the *genotype space*, such a set of image properties is *not* the genotype space. Such image properties are *measures*

of images that will be used to help measure their impressiveness, and do not specify particular images themselves. For this experiment, eight features are chosen to capture the space of image properties, motivated by their simplicity and alignment with human recognition:

Brightness. An average of all pixel values in the picture yields a measure of a picture's brightness.

BZip2 compression. The compressibility of the image by the BZip2 algorithm gives an estimate of the picture's visual complexity.

Wavelet compression. This measure describes how compressible the image is after a wavelet transformation by counting how many coefficients are necessary to explain 95% of the image's brightness. Wavelet compression offers an alternate perspective to BZip2 on complexity.

Color variety. The standard deviation statistic is calculated over of all pixel values in a picture, giving a measure of how widely pixel values are distributed.

X-axis symmetry. This simple measure of symmetry is calculated by taking the average pixel similarity between pixels reflected over the X-axis.

Y-axis symmetry. The same measure as above is instead applied to the Y-axis.

Choppiness. The discontinuity of local neighborhoods of pixels is estimated by this measure. It is calculated as the average standard deviation of pixels over all 5x5 windows within the picture.

While the idea of impressiveness does not depend on this particular choice of picture properties, the general motivation is that such a set can facilitate aligning impressiveness with pictures visually appreciated by humans. Furthermore, they enable the evolution of impressive pictures because the trade-offs between various properties are difficult to achieve. For example, maximizing one compression measure while minimizing the other requires exploiting the differences between the underlying compression algorithms.

However, an interesting question is how to evolve pictures with such impressive properties. To do so a means of representing and evolving pictures is necessary. While there are many different representations for pictures, a well-validated method is to apply the NeuroEvolution of Augmenting Topologies (NEAT; Stanley and Miikkulainen 2002, 2004) algorithm to pictures represented by compositional pattern producing networks (CPPNs; Stanley 2007), as in Picbreeder (Secretan et al., 2011; Stanley, 2007). While the NEAT method was originally developed to evolve artificial neural networks (ANNs) to solve difficult control tasks (Stanley and Miikkulainen, 2002, 2004), it is easily adapted to evolving CPPNs because they are similar in structure to

ANNs. Also, NEAT is well-suited to evolving impressive pictures because it can *complexify* CPPN topology into diverse species over generations, leading to increasingly sophisticated pictures.

In effect, CPPNs are neural networks extended to contain a variety of specially-chosen activation functions. The CPPNs in this paper take x, y coordinates as input and output the pixel brightness at that location. They facilitate images with regularities through activation functions with regular properties. For example, a Gaussian activation function by virtue of its symmetry can induce symmetric pictures and a sine function can induce pictures with elements of repetition. In this way, evolving CPPNs with NEAT can result in increasingly sophisticated images with appreciable regularities (as seen in Picbreeder; Secretan et al. 2011), which aligns well with the motivation for the experiment in this paper. Importantly, all of the experimental setups that follow apply NEAT with the same settings to evolve CPPNs; only the *reward scheme* is varied between them.

Varying the reward scheme in this way facilitates exploring the question of what type of evolutionary reward scheme is appropriate to guide this kind of open-ended search. Most approaches in EC apply objective-driven fitness functions. Yet in the huge space of potential pictures there are no inherent notions of better or worse, which usually underlies the traditional fitness-based search paradigm.

Thus with open-ended evolution in mind, a promising approach is to reward *exploring* the space of pictures through novelty search. That is, a picture is rewarded proportionally to how novel it is, i.e. how different it is from previously encountered pictures with respect to the eight picture properties (which are each scaled between 0 and 1 so that they are equally weighted). The idea is that over time as the easiest to reach points in this space are exhausted, evolution will be driven into interesting trade-offs and areas of the space that are increasingly difficult to reach. That is, novelty search may be driven to find impressive pictures. However, this sort of search has no ultimate objective other than to continually uncover new varieties of pictures and thus aligns well with the idea of open-ended evolution.

Two alternate reward schemes are also considered for comparison. First, a random search is implemented in which pictures are rewarded random fitness. The idea is to explore whether a random search, which is also open-ended in some sense because it does not attempt to prune out any possibilities from search, can also discover impressive artifacts through drift combined with NEAT's drive to complexify over time. Second, a fitness-based search is considered in which the explicit objective for each run is to re-evolve one of the rarest pictures discovered by novelty search. That is, the fitness function is to minimize distance among the salient properties (explained in the next section) from an evolved picture to the target picture. The hypothesis is that impressive artifacts may also be *deceptive* as targets and thus hard

to reach directly. If this hypothesis is true then an objective-based search to recreate such rarity may often fail to discover pictures as impressive as the target.

In this way, one aim of the experiment is to discover whether the proposed measures can make meaningful distinctions between variations in reward scheme that would naturally be expected to impact the dynamics of impressiveness. The measure’s ability to make such distinctions may predict its applicability to other artificial life experiments.

Experimental Parameters

For each reward scheme 40 independent runs were conducted that ran for 500 generations each with a population size of 250. Evolved pictures were 64x64 pixels. Unlike in Picbreeder, colors in the pictures were limited to grayscale for simplicity. The dynamic threshold for adding pictures to the novelty archive was initialized to 0.5. The weight mutation power was 1.0, the chance for adding a new node was 0.05, and the chance for adding a connection was 0.1.

Results

To analyze the products of the picture evolution system, the two heuristics of rarity and re-creation effort were fitted to the domain and applied, which the next section discusses.

Recognizer Based on Rarity

To estimate how rare combinations of various values of the eight measured image properties were, ten million random CPPNs of various complexities were sampled and their properties measured. Histograms were constructed (with bins with width 0.05) for each combination of properties to estimate their joint probabilities (e.g. one such histogram would bin based on three dimensions: levels of x symmetry, wavelet compressibility, and brightness). In this way, the rarity within the space of random CPPNs of certain combinations of properties can be approximated.

To model recognition of an image’s most salient features, a recognition algorithm was created that when applied to a picture would return the rarest combination of properties (e.g. the most improbable combination of properties for a particular picture might be a x-symmetry of level 0.3 and BZip2 compressibility of 0.6). In other words, the recognizer returns a summary of what is most unique about a picture, and how rare such an abstract description is (i.e. how often it occurs among randomly sampled CPPNs). Formally, the rarity of an evolved artifact is defined as $-\log(p(\alpha))$, where α is the set of salient features and $p(x)$ is a function that estimates the probability of such features occurring by chance, i.e. the probability returned by the recognizer.

In particular, the recognizer is a greedy algorithm that iterates over each combination of k features searching for the most improbable among possible combinations, starting with $k = 1$ and increasing incrementally. Because joint

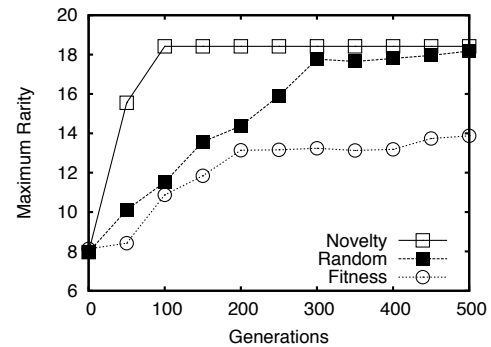


Figure 1: **Rarity of evolved images.** The maximum rarity (i.e. how infrequently similar pictures occur) of pictures from novelty search, random search, and fitness-based search is shown over 500 generations of evolution averaged over 40 independent runs. Note that a combination of image properties not present in any of the sampled CPPNs will receive a rarity of 18.4 ($2^{18.4} \approx 10,000,000$), the line to which novelty search quickly converges.

probabilities can only decrease when adding additional features, a control is added to ensure that adding a new feature increases rarity by at least 10 times the a priori assumption of a uniform distribution; otherwise the algorithm would terminate and return the most rare combination found so far. In this way, only the most unique properties would be considered that significantly contribute to rarity, i.e. this constraint acts as a filter to ensure concise descriptions of artifacts.

After the histograms are computed from the random CPPN samples, the recognizer algorithm is computationally inexpensive and can thus be applied to all evolved artifacts from each run at 50 generation intervals. Figure 1 shows the results of averaging the most rare picture discovered by a particular run over generations as measured by the recognizer. The main result is that rarity is able to distinguish between the different reward schemes. Novelty search is most driven towards rarity while random search more slowly discovers rarer artifacts over time (the difference is significant from generation 50 until generation 250; Student’s t-test; $p < 0.001$). Novelty search also discovers significantly more rare artifacts than fitness-based search from generation 50 onwards (Student’s t-test; $p < 0.001$). Interestingly, directly searching to recreate rare pictures with fitness-based search often fails due to deception.

A selection of such rare pictures found by novelty search is shown in figure 2. To aid in interpretation the combination of properties that justifies each image’s rarity is returned by the observer. That is, it is possible to provide objective evidence for what is impressive about these images, instead of relying on subjective assessment as is often necessary when describing the results of an artificial life simulation. For example, picture 2a is highly compressible by BZip2 yet relatively incompressible by the wavelet algorithm, and has a low average pixel value. The result is impressive because



Figure 2: **Selection of rare pictures.** Each of these pictures discovered by novelty search was evaluated as rare by the observer because it has a combination of properties that rarely co-occur within the space of pictures.

these settings mutually conflict; generally an image is either compressible or not compressible, and incompressibility is more easily attained through wildly fluctuating pixel value (which would yield a higher average). It is also possible to learn about an encoding through observing rare artifacts: Figure 2c is rare because it is highly asymmetric along both the x and y axes and such rigidly rectangular asymmetry is not a natural bias of CPPNs (nor is it of DNA in nature).

To investigate the results of the picture evolution experiment further, the next section describes applying a more rigorous heuristic of impressiveness.

Re-creation Effort

While rarity provides one heuristic for the impressiveness of an artifact, not all that is rare is difficult to achieve. Thus conceivably the rare artifacts discovered by novelty search may require little effort to recreate, which would undermine their impressiveness.

Therefore, as a more rigorous heuristic of impressiveness, the effort required to recreate artifacts was estimated. The basic idea is to measure how much effort on average it takes to recreate a similar artifact from scratch. First, because it is computationally expensive to calculate, only the most rare picture was sampled across all 40 runs of all methods at 100 generation intervals. For each sampled picture, the observer described in the previous section derived the most rare combination of properties. Next, for each set of such observed properties five independent runs of NEAT were instantiated with those properties as an explicit objective (i.e. the fitness function was to minimize distance between the most unique properties of the target image and a candidate solution image). Each run terminated if unsuccessful after 50,000 evaluations, or if the image *properties* were successfully recreated. The average number of evaluations required to evolve an image that would fall into the same histogram bin (i.e. allowing for error of 0.05 in any given property) was then recorded as an estimate of the effort required to recreate a similar picture.

Figure 3 shows these results for all three variants, which reinforce the results from measuring rarity in the previous section by distinguishing the reward schemes in the same order. In particular, novelty search is distinguished from ran-

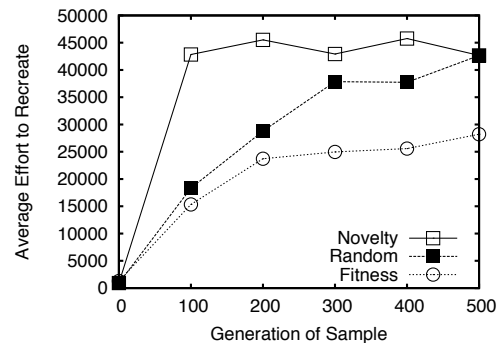


Figure 3: **Effort to recreate image properties.** The average effort (i.e. the number of evaluations) necessary on average to recreate the rarest images (with fitness-based search) sampled at 100 generation intervals from each run of novelty search, random search, and fitness-based search is shown. Note that the measure has a ceiling of 50,000 evaluations, which may mask continuing growth of re-creation effort for both novelty search and random search.

dom search for generations 100 and 200, and from fitness-based search for all generations after zero (Student's t-test; $p < 0.001$). It is interesting that random search demonstrates a drive towards impressiveness (which may result from NEAT's complexification mechanism), although novelty search most quickly evolves artifacts that exceed the upper extreme of the test's range (50,000 evaluations).

Additionally, a significant correlation (0.673) was measured between paired samples of rarity and re-creation effort ($p < 0.0001$; Kendall's tau coefficient), indicating that the two heuristics are strongly related, which supports their derivation from the same definition.

Discussion

From a practical perspective the definition and heuristics of impressiveness introduced in this paper facilitate making distinctions among variations of evolutionary systems and providing objective statements about their products. Novelty search, which is designed explicitly to achieve open-ended exploration, climbs the ladder of impressiveness most steeply, as would be intuitively expected. However, on a deeper level impressiveness yields an alternate perspective on the goals of open-ended evolution.

That is, perhaps meeting the challenge of unbounded open-endedness, which is often assumed to correlate with intuitions about natural evolution's greatness, is instead a necessary but not sufficient condition to yield increasingly impressive products. In other words, increasing impressiveness may be a more inherently meaningful goal than open-endedness alone insofar as it more deeply abstracts what we appreciate about natural evolution: its impressive products.

Furthermore, the results in this paper and prior work with non-objective search processes (such as novelty search and Picbreeder) suggest that objective-based search is deceived by increasingly ambitious or impressive objectives (Lehman and Stanley, 2011a,b; Woolley and Stanley, 2011). Thus, an interesting possibility is that open-endedness may be important to evolving increasingly impressive artifacts solely to circumvent deception. That is, seeking impressiveness convergently may be fruitless because of the inherent difficulty in predicting a priori what paths through any search space will lead to great achievement. Such a possibility hints at a potential deeper understanding of open-ended creativity.

Future work will investigate the hypothesis that systems previously passing the evolutionary activity statistics tests will not exhibit unbounded impressiveness, highlighting where the two measures may differ.

Conclusion

Motivated by the possible gap between open-endedness and impressiveness in some artificial life simulations, this paper introduced the idea of quantifying the impressiveness of evolved artifacts. Heuristic measures of impressiveness derived from a novel definition were applied to an open-ended picture evolution system to characterize the effect of different reward schemes on impressiveness and to examine individual evolved products. The conclusion is that impressiveness is a new tool for investigating the products of open-ended systems that presents an alternate perspective on the goals of open-ended evolution.

Acknowledgements

This research was supported by DARPA and ARO through DARPA grant N11AP20003 (Computer Science Study Group Phase 3) and US Army Research Office grant Award No. W911NF-11-1-0489. This paper does not necessarily reflect the position or policy of the government, and no official endorsement should be inferred.

References

- Bedau, M., Snyder, E., Brown, C. T., and Packard, N. H. (1997). A comparison of evolutionary activity in artificial evolving systems and in the biosphere. In Husbands, P. and Harvey, I., editors, *Proceedings Of The Fourth European Conference on Artificial Life*, pages 125–134. MIT Press.
- Bedau, M. A., Snyder, E., and Packard, N. H. (1998). A classification of longterm evolutionary dynamics. In Adami, C., Belew, R., Kitano, H., and Taylor, C., editors, *Proceedings of Artificial Life VI*, pages 228–237, Cambridge, MA. MIT Press.
- Channon, A. (2001). Passing the alife test: Activity statistics classify evolution in geb as unbounded. In *Proceedings of the European Conference on Artificial Life (ECAL-2001)*. Springer.
- Channon, A. D. and Damper, R. I. (2000). Towards the evolutionary emergence of increasingly complex advantageous behaviours. *International Journal of Systems Science*, 31(7):843–860.
- Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection or the Preservation of Favored Races in the Struggle for Life*. Murray, London.
- Gasarch, W. (2002). The P=? NP poll. *Sigact News*, 33(2):34–47.
- Geng, L. and Hamilton, H. (2006). Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR)*, 38(3):9.
- Kelly, K. (2010). *What technology wants*. Viking Press.
- Lehman, J. and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In Bullock, S., Noble, J., Watson, R., and Bedau, M., editors, *Proceedings of the Eleventh International Conference on Artificial Life (ALIFE XI)*, Cambridge, MA. MIT Press.
- Lehman, J. and Stanley, K. O. (2011a). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*.
- Lehman, J. and Stanley, K. O. (2011b). Novelty search and the problem with objectives. In *Genetic Programming in Theory and Practice IX (GPTP 2011)*, chapter 3, pages 37–56. Springer.
- Maley, C. C. (1999). Four steps toward open-ended evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-1999)*, volume 2, pages 1336–1343, Orlando, Florida, USA. IEEE Press.
- Nehaniv, C. (2000). Measuring evolvability as the rate of complexity increase. In Maley, C. and Boudreau, E., editors, *Artificial Life VII Workshop Proceedings*, pages 55–57.
- Neill, A. and Ridley, A. (1995). *The philosophy of art: readings ancient and modern*, pages 98–239. McGraw-Hill.
- Reeves, C. (2000). Fitness landscapes and evolutionary algorithms. In *Artificial Evolution*, pages 3–20. Springer.
- Schmidhuber, J. (2009). Driven by compression progress: A simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. *Anticipatory Behavior in Adaptive Learning Systems*, pages 48–76.
- Secretan, J., Beato, N., D'Ambrosio, D., Rodriguez, A., Campbell, A., Folsom-Kovarik, J., and Stanley, K. (2011). Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, 19(3):373–403.
- Standish, R. (2003). Open-ended artificial evolution. *International Journal of Computational Intelligence and Applications*, 3(167).
- Stanley, K. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162.
- Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127.
- Stanley, K. O. and Miikkulainen, R. (2004). Competitive coevolution through evolutionary complexification. 21:63–100.
- Taylor, T. and Hallam, J. (1998). Replaying the tape: An investigation into the role of contingency in evolution. In Taylor, C., Langton, C., and Kitano, H., editors, *Proceedings of Artificial Life VI*, pages 256–265.
- Woolley, B. G. and Stanley, K. O. (2011). On the deleterious effects of a priori objectives on evolution and representation. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2011)*. ACM.

Architectures for Self-reproduction: Abstractions, Realisations and a Research Program

Barry McMullin

Dublin City University, Ireland
barry.mcmullin@dcu.ie

Abstract

It is well recognised that von Neumann's seminal abstraction of machine self-reproduction can be related to the reality of biological self-reproduction — albeit only in very general terms. On the other hand, the most thoroughly studied artificial evolutionary systems, incorporating meaningful self-reproduction, are the coreworld systems such as *Tierra*, *Avida* etc.; and these, in general, rely on a purely “self-inspection” mode of reproduction (or, more simply, “replication”). To the extent that the latter has any direct biological analog it would appear to be with molecular level reproduction and evolution in the hypothesised RNA-world. In this paper I review the details and distinctions between these modes of reproduction. I indicate how the abstract von Neumann architecture can, in fact, be readily realised in coreworld systems; and outline the research program that flows from this. Finally I attempt to make more precise the resulting analogies with molecular biology, at least up to the (prokaryotic) cell level.

Background: von Neumann's Problem

As is well known, in the late 1940s and early 1950s John von Neumann conducted an investigation into problems of understanding the evolutionary growth of complexity (Burks, 1966; McMullin, 2000). In particular, he wondered how it could be possible for a machine, of a given level of complexity, to construct an offspring machine of greater complexity than itself. *Prima facie*, from an engineering point of view, this seems like a paradox: surely any machine capable of constructing other machines must, in some sense, already contain the design of those machines within itself; and therefore must already be more complex than any such offspring machine. And yet, if the theory of biological evolution (at least of evolutionary descent from one, or a small number, of primordial ancestor species) is correct, and if biological organisms are some (possibly very special) kind of “machine”, then this sort of constructive increase in complexity must not just be possible, but must happen repeatedly, indeed almost continuously, over evolutionary time. As a special (edge) case, even without considering evolutionary *growth* of complexity, viewing organisms as machines at all, and recognising their universal capacity for *self-reproduction*, implies that essentially arbitrarily complex machines can construct

offspring of at least *equal* complexity to themselves. So, *inter alia*, von Neumann wondered how it can be that arbitrarily complex machines can be capable of such self-reproduction.

The formulation (and thus solution) of these problems may seem to hinge critically on what we mean by “complexity”; but for the immediate purposes of this paper it will suffice to adopt von Neumann's own, vague and qualitative, definition that complexity means the ability to “... do very difficult and involved things” (Burks, 1966, p. 78). Specifically, the machines under consideration must exist in a universe in which they can “do” more or less “difficult and involved” things, ideally with no obvious upper bound.

With this starting point, von Neumann proceeded to formulate a completely general and abstract machine architecture whereby such machines could:

- be arbitrarily complex (i.e., the set of such machines would span whatever range of complexity is possible at all within their universe),
- be capable of self-reproduction,
- be capable of undergoing spontaneous “mutation”, giving rise to offspring which are different in kind, but this difference is retained through further cycles of reproduction — i.e., the differences “breed true”,
- and where the entire set of such machines is connected under mutation.

That is, starting with an arbitrarily simple machine having such an architecture, there would exist sequences of possible mutations leading to machines of the highest complexity possible in the particular universe, where all of these machines share the same architecture and all are also capable of self-reproduction.¹

¹The bare existence of such sequences does not guarantee that any would ever be followed. That would depend on quite separate factors — most critically, what ecological and selectional interactions arise between mutationally distinct machine lineages. However, such questions will fall outside our immediate scope here.

Having formulated the general architecture, von Neumann went on to consider its realisation in particular “model” universes. He first imagined an abstract, but still quite physically motivated, “kinematic automaton” universe — not unlike a modern “physics engine” world such as commonly applied in certain forms of computer gaming and animation. His most detailed elaboration was in the case of a two dimensional, homogeneous, “tessellation automaton”, or “cellular automaton” (CA) universe as it would now be called. He did successfully demonstrate, in this particular CA universe, the essential detailed design of a particular “seed” machine that had his abstract architecture (Burks, 1966). This was sufficient to establish the principle that there could also exist, in this universe, arbitrarily complex machines which would also be logically capable of self-reproduction, and that this entire (infinite) set of self-reproducing machines would be connected under mutation.

The Abstract von Neumann Architecture

The abstract architecture described by von Neumann consists of a complete machine, denoted M , having a highest level decomposition into an active, functional, component, labelled P , and a relatively passive component labelled G . We write $M = (P + G)$.

G consists of a linear chain of sub-components. Each sub-component can be chosen from some finite set of possible component types. We assume there are at least two such distinct types or configurations. G can be of arbitrary length, and the different component types must be “compatible” in the sense that the chain can be constructed with an arbitrary sequence of these allowed types. G can then be thought of as roughly analogous to the *tape* of a Turing machine, effectively acting as an information storage system. The particular choice of component at a particular position corresponds to, or represents, the discrete “symbol” recorded in a single “square” of a Turing machine tape. G has no activity or functionality in its own right: it just retains the specific sequence of its components, which is to say, the specific information stored in it.²

P consists of four distinct, but functional and interacting, “sub-machines” or “sub-assemblies”:

A : This is a “general constructive automaton” or “programmable constructor”.³ By hypothesis, A can interpret the information stored in the chain G (in the specific sequence of its sub-components) as representing the

description of some essentially arbitrary configuration or assembly of whatever types of machine components are made available in the particular universe. That is to say, A can construct any (largely) arbitrary target (sub-)machine. For the moment we label the latter as X : so we can say that G contains a description of some machine X (relative to A ’s particular interpretation). We denote this by saying that G stores the information $\phi(X)$ or simply $G = \phi(X)$ (where the function ϕ is determined by the detailed design of A). Once activated, A will scan G and proceed to construct an instance of X .⁴

B : This is a “general tape copier”. Once activated, it will scan G and construct another, separate, chain, with exactly the same sequence of sub-components (i.e., storing the same “information”).

C : This is a “controller”, which primarily acts to sequence the operations of other sub-assemblies of P . It operates in a cyclical manner. It activates A and B in turn (the order does not matter in general), or possibly even concurrently (provided they will not interfere with each other — in particular in their access to G). It then connects together the two newly constructed sub-assemblies — the sub-machine X and the copy of G — forming $(X + G)$ and releases this into the surrounding environment, as a new, discrete and separate, machine. In general this releasing step is also thought of as incorporating an initial “activation” of the new, offspring, machine. C then starts the cycle again from the beginning, and repeats this indefinitely (for as long as M remains functional or “alive”).

D : This denotes an arbitrary assembly of the components allowed within the universe, i.e., it effectively represents an arbitrary (sub-)machine in its own right, with arbitrary functionality (and thus “complexity”). D can be supposed to operate autonomously of, and indeed concurrently with, the other sub-assemblies, with the one constraint that it must not compromise or interfere with the functionality already described for A , B and C . If necessary, it can be assumed that in addition to sequencing A and B , C also co-ordinates with, or even controls, D to assure this independence of operation.⁵

⁴In the general case, this process may “fail” for various reasons. G may not be correctly structured as a “tape” at all (it may have the wrong morphology — not be a linear chain — or have incorporated components not recognised as denoting symbols etc.). There may be configurations for G that are correctly structured (as sequences of symbols) but which do not describe any machine relative to A . There may be possible machines X which A cannot construct (i.e., for which there is no corresponding description $\phi(X)$ relative to A). However, I will not consider these issues further here.

⁵von Neumann referred to D as “ancillary” machinery, but this is somewhat misleading. In the general case, D may represent the great bulk of the physical constitution of the overall machine M and, in any case, determines any and all of its *distinctive* functionality (i.e., over and above self-reproduction *per se*).

²As an aside, for our particular purposes here (and unlike a conventional Turing machine tape) it is not essential that each individual “square” of G be capable of being “rewritten” with an arbitrarily different (allowed) symbol. It is sufficient if each position can be made to hold an arbitrary symbol at the time the chain is initially constructed. In this sense, G is more like a “read only memory” than a conventional, Turing tape-style, “read write memory”.

³ A is commonly referred to as a “universal constructor”. However, that usage is somewhat problematic, and I will avoid it here.

In summary then, a generic behaviour of any machine having this architecture is to repeatedly construct “offspring” machines of the form $(X + G)$, which is to say $(X + \phi(X))$, for arbitrary X . But since X is arbitrary, we can now take the final step of *stipulating* that $X = P$.⁶ Thus, in designing the machine M we would first design P as already described above and then deliberately arrange that the information recorded into G is just that sequence which, under the function $\phi()$, represents P itself, i.e., $G \triangleq \phi(P)$. With this identification, we have $M = (P + \phi(P))$ and the offspring of M is precisely:

$$\begin{aligned}(X + G) &= (X + \phi(X)) \\ &= (P + \phi(P)) \\ &= M\end{aligned}$$

i.e., M is now self-reproducing. This is true of the entire family of machines represented by M , having arbitrary (and thus arbitrarily complex) sub-systems (“ancillary machinery”) D .

Mutation

As already noted, an essential aspect of von Neumann’s problem was that he wanted to understand or model a general mechanism for *mutational change*, that is, spontaneous or accidental modifications of a (self-reproducing) machine’s structure that would nonetheless *breed true*. With respect to his abstract architecture we will now systematically explore the possibilities.

First consider modifications to the structure of $P = A + B + C + D$. In the case of changes to any of A , B or C the expected outcome is simply that the reproductive functionality will be broken — the machine will cease to construct any further offspring, or any offspring it does construct will be malformed, probably without function, and certainly no longer identical to the parent. So such changes cannot breed true.

In the case of an alteration to D , say from D to D' , then, as long as this does not compromise the ongoing operation of $A + B + C$, the machine will continue to produce offspring. But since the P part of these offspring is constructed by A interpreting (or “decoding”) $G = \phi(P)$, these offspring will revert to the form $P + \phi(P) = (A + B + C + D) + \phi(A + B + C + D)$ and not the altered form $P' + \phi(P) = (A + B + C + D') + \phi(A + B + C + D)$ that the parent now has. So again, such changes will not breed true.

⁶There are subtleties here. In particular, it must be the case that each particular P (i.e., defined by $A + B + C$ combined with a particular D) is indeed constructable by A . As already noted, this is not necessarily the case in general; but for my purposes here I shall simply assume that this condition can be satisfied with “sufficient” generality (i.e., for a sufficiently wide definition of D).

Now let us consider a modification to the structure of G , changing it to G' . In the first instance we now expect that the active machinery of M (P) will continue in operation, so it will continue to produce offspring. Further, since the B component simply copies whatever chain or information sequence it is presented with, this offspring will contain G' rather than G ; but this will not be the only difference in the offspring. In general the offspring P part will also be different, as it now results from A decoding G' rather than G . Assuming that, relative to A , G' codes for any functional machinery at all, we can denote this as P' where $G' = \phi(P')$. So now the produced offspring have the form $M' = (P' + \phi(P'))$. The subsequent behaviour of these offspring — and in particular whether the change will now breed true — will depend on the exact nature of the differences between P and P' .

Note that even though (by hypothesis) we are considering only a local (“point”) change from G to G' , it does not follow that the change from P to P' will be similarly localised: this depends entirely on the nature of the decoding function ($\phi^{-1}()$) implemented by A . In the general case, $P' = \phi^{-1}(G')$ might be arbitrarily different from P . Nonetheless, for our current purposes we will suppose that the effect is at least approximately local; in particular, let us consider the case where the difference between P and P' is localised to just one of the components A , B , C or D .

The simplest case to understand is where only D is affected, changing to D' , i.e., $P' = A + B + C + D'$. In this case, assuming that D' does not interfere with the operation of A , B or C , then $M' = (A + B + C + D') + \phi(A + B + C + D')$ has exactly the same abstract architecture as the original M and therefore will successfully self-reproduce. Thus, any changes of this sort — changes in G that affect only the D part and function of the offspring — will indeed subsequently breed true. Such changes fully qualify as *heritable mutations* in the normal biological sense. Further, if it so happens (as it sometimes may) that D' is even slightly more complex (more complicated in its behaviour) than D , then this will be an example of an incremental evolutionary growth in machine complexity. Ultimately the entire set of machines of the form $M' = (A + B + C + D') + \phi(A + B + C + D')$ for arbitrary D' are all individually self-reproducing and are fully connected together through this network of possible heritable mutations — spontaneous variations in G that do not affect A , B or C in the offspring.

At this point we can recognise that decomposition of M into P and G is closely analogous to the biological decomposition of an (individual) organism into its *phenome* and *genome* respectively. Broadly speaking, modifications to the phenome may or may not impair the ability of the organism to reproduce; but if it *can* still reproduce then the offspring will not inherit this purely phenotypic change to the parent. Conversely, at least some genotypic changes to the parent

will both be expressed in the offspring phenome and subsequently breed true. And indeed if, as previously mentioned, the D component forms the bulk of the machine M , and, correspondingly, the bulk of G codes for D , then we can expect that the bulk of possible changes to G will, if they result in viable offspring at all, fall into this category of giving rise to heritable mutations. Further, we can now recognise the “decoding” function ϕ^{-1} , implemented by A , as corresponding to what would be called a *genotype-phenotype* mapping in biology. For notational convenience below, we will also denote this mapping by $\psi \triangleq \phi^{-1}$.

But let us now return to the final remaining cases of possible machine modifications; namely a change to G to G' (i.e., still a “genotypic” change) where this results in a change of (one of) A , B or C in the offspring. At first sight, one is inclined to suppose that the earlier analysis of *direct* changes to A , B or C in the parent machine can be immediately re-applied to this (first generation) offspring, and conclude that all such mutant offspring will in fact be sterile (so that these cases will also have no evolutionary significance). And indeed, in von Neumann’s original presentation that is exactly the position he adopted. However, in fact, this overlooks additional possibilities which do merit deeper consideration. Viable changes to B or C should surely not be absolutely ruled out; but for our particular purposes here we will choose to focus specifically on the role of changes localised to subsystem A , the programmable constructor.

Let us consider then the behaviour of a first generation offspring machine of the form $M' = (A' + B + C + D) + \phi(A' + B + C + D)$. If A' is completely broken, then, indeed, M' will be sterile, as von Neumann supposed. But what if A' is still essentially functional in the sense that it does “decode” $G' = \phi(A' + B + C + D)$ to produce the component P' of its (now second generation) offspring; *but* the exact decoding function implemented by A' has changed — it is no longer $\psi()$ but some more or less modified function $\psi'()$?

In this case the outcome depends critically on the detailed behaviour of this modified decoding function; indeed, it depends on the behaviour of this function precisely for the particular element of its domain represented by $G' = \phi(A' + B + C + D)$. Probably the most common case will be that this decodes as some arbitrary $P' = \psi'(\phi(A' + B + C + D))$. No general further analysis of that case is possible, but we can reasonably expect that that (second generation) offspring will be largely or wholly non-functional i.e., we again conclude that M' is sterile.

But there is still the logical possibility that, even though $\psi \neq \psi'$ (by hypothesis) we might still have $\psi'(\phi(A' + B + C + D)) = (A' + B + C + D)$, i.e., for that particular G' the two functions might still co-incide. We would say that ψ' (or A') is *backwards compatible* with ψ (or A) for that particular value of their common domain. In that (special, but conceivable) case, the machine M' can be equally well represented as having the structure $A' + B + C + D +$

$\phi'(A' + B + C + D)$. By the logic of the abstract architecture, this is indeed self-reproducing, so we again have an example of a heritable mutation; but of a quite different order to the “normal” case of a simple change affecting D' . In fact, this would correspond to a *mutation in the genotype-phenotype mapping itself*. In one mutational step, we would have moved from exploring the space of machines $M = (A + B + C + D) + \phi(A + B + C + D)$, with arbitrary D and all sharing a common genotype-phenotype map ψ defined by A , to exploring a different space of machines $M' = (A' + B + C + D) + \phi'(A' + B + C + D)$, still with arbitrary D , but now sharing a different genotype-phenotype map ψ' . This may not seem like a big change at first sight — as the space of machines D , which putatively defines all of the “interesting” variation in complexity, may still be essentially identical in both cases. But, the topology of the mutational connections — and thus the dynamics of evolutionary change, and the ultimate complexity which emerges — may be radically different.

The very least we might conclude is that any system that has this possibility for evolutionary modification of the genotype-phenotype map has, at the very least, some additional degrees of evolutionary freedom that would not otherwise be available. This will depend, of course, on the space of “possible” decoding functions $\psi()$ and on the subset of these that might be mutationally accessible from any particular starting point (the particular $\psi()$ implemented by some particular, initial, seed, machine). It seems very difficult to make any general statement about this; but one particular case would be where the potential machinery represented by A is capable of incorporating, as part of the decoding process, any arbitrary Turing computation (i.e., the available configurations for A encompass the flexibility of universal computation). This would at least guarantee that the space of possible genotype-phenotype mappings (and corresponding evolutionary dynamics) spans a very wide range of possibilities. However, even this still leaves completely open the question of how richly connected this mutational space might be; i.e., how common would be the “backwards compatible” mutations that actually allow ψ to successfully mutate?

Finally for this discussion we should note again that this possibility of a mutable genotype-phenotype mapping relies on a peculiarly tangled loop of inter-dependency. For self-reproduction to work in von Neumann’s architecture, the programmable constructor (A) must always be described or encoded into the genome under a mapping ($\phi()$) which precisely inverts the mapping that that constructor itself physically realises ($\psi()$), at least for that one particular description represented by genome G . Howard Pattee in particular has especially drawn attention to this aspect of von Neumann’s architecture, and distinguished it with the term *semantic closure* (Pattee, 1982).

Self-Reproduction in Coreworlds

As noted, the most detailed proposal for an artificial realisation of the von Neumann self-reproduction architecture was in his two dimensional CA universe. Von Neumann himself, having set aside this work in the form of a planned, but unfinished manuscript, did not have the opportunity to return to it before his untimely death in 1957. In any case, the detail and required scale of the model was such that it is only very recently that it has become technically feasible to build a practical implementation⁷. More importantly, and quite independently of scaling issues, that particular realisation was never expected or intended to support investigation of any substantive evolutionary processes, because the machines are intrinsically fragile with no ability to interact (even with their own offspring) without causing catastrophic breakdown in organisation (McMullin, 2000).

By contrast, so-called *coreworld* systems do allow practical investigation of relatively large populations of self-reproducing agents over sufficient time to allow significant evolutionary change. These agents can be conceived as as small machine code programs, each occupying a block of allocated memory and executed by a dedicated processor (CPU). Memory blocks and CPUs are managed and allocated from a common pool.

The roots of the coreworld concept can be traced back to Nils Barricelli, who was invited to visit the Institute for Advanced Studies (IAS) in Princeton by von Neumann in the early 1950s (Barricelli, 1957).⁸ However, the more recent and most extensively studied systems of this type are *Tierra* (Ray, 1992, 1994) and *Avida* (Adami and Brown, 1994; Adami, 1998).

While these systems do involve self-reproduction of a sort, to date this has not been implemented using the von Neumann architecture. Instead, reproduction is achieved by direct “self-inspection”.⁹ That is, these universes are designed in such a way that machines (realised as software agents or processes) of arbitrary complexity (within the context of the particular universe) can successfully examine their own detailed internal structure (their own memory image) without disruption; and can copy this structure into a newly allocated memory block, allocate a separate CPU, and then release an essentially identical offspring machine into the (shared) environment. In terms of the von Neumann architecture it is as if the G and P components are com-

bined in one single, active, structure which also serves as its own description; accordingly, it need only be copied — no “decoding” step is needed as the copied structure is already directly constructed in the required “functional” form. But viewed as P we can still decompose it into a part concerned with self-inspection/copying (corresponding to von Neumann’s B) and a part corresponding to all machinery not directly concerned with reproduction (von Neumann’s D). There is no A or C , and no separate G . Instead, we have just $G \equiv P = (B + D)$

Heritable mutations are still perfectly possible. Any modification anywhere in the memory image which does not impair the inspection (reproduction) functionality (i.e., almost any change to part D , and typically at least some changes to part A) will result in an offspring which is still capable of self-reproduction, preserving that modification — i.e., the modified/mutated machine will breed true.

Accordingly, in this architecture there is no decomposition into a separate “phenome” and “genome”. In effect, the whole machine (or at least its tape/memory image) is simultaneously both phenome and genome.

Despite this seemingly radical simplification, a family of such self-inspectors (defined by a specific B and arbitrary D) still meets all the desiderata to address von Neumann’s problem as it was expressed earlier.

Now this kind of pure self-inspection architecture need not be feasible in general. Thus, in von Neumann’s own CA based model universe, it is not possible for an arbitrary machine to completely inspect its own internal structure without disruption — it would effectively have to disassemble (i.e., destroy) itself in the process. Similarly, in the physical-chemical world of real biology, there are serious limitations to the possibilities for self-inspection. In such universes, the von Neumann architecture, with its separation of the functional, dynamic, phenome from a static, linear (completely open to non-destructive inspection) genome clearly enables a qualitatively richer set of self-reproducing machine configurations, and thus a qualitatively richer potential for evolutionary exploration. But conversely, in universes such as coreworlds, which have been specifically engineered to support comprehensive, non-destructive, inspection of arbitrary machine configurations, it would seem that all conceivable evolutionary phenomenology must already be available via machines with the self-inspection architecture; so even if machines with the full von Neumann architecture *could* be realised in such universes, there would arguably be no point or interest in doing so.

Nonetheless, there are grounds for suggesting that this issue should not yet be completely closed. In particular, note that if we restrict attention to self-inspectors, not only is there no decomposition into genome (G) and phenome (P); there is also no property of semantic closure (in Pattee’s sense) and no *mutable mapping from genotype space to phenotype space*. Given the discussion of the previous

⁷It is reported that “... in 2008, the hashlife algorithm was extended to support the 29-state and 32-state rulesets ... [and] On a modern desktop PC, replication now takes only a few minutes” (Wikipedia contributors, 2012).

⁸Barricelli actually did his earliest work of this type on the “IAS Machine”, the stored programme electronic digital computer built at Princeton to von Neumann’s original design and under his direction.

⁹We focus here on coreworld systems; but note that the self-inspection mechanism has been proposed in other frameworks also (e.g., Laing, 1977; Morita and Imai, 1996).

section we would like to hold open at least the possibility that even though the von Neumann architecture is not *necessary* for “general purpose” self-reproduction in coreworlds, nonetheless it may be *useful*; it may give rise to evolutionary dynamics and phenomenology (including exploration or complexification of the genotype-phenotype mapping) that would not otherwise arise.

So, can we say whether it is possible in principle to embed agents having the von Neumann architecture in coreworld systems?

The general, in principle, answer is “yes”. A coreworld already provides facilities for a machine (software agent) to allocate an additional memory block, configure it (write into it) as desired, allocate a CPU to start execution on that memory block, and release this functioning assembly into the environment as a separate, autonomous, agent. As it is possible for an agent to inspect and copy its entire memory image into the offspring memory block, then it should surely be possible to copy only some part of the image (to be identified as G). The remaining part of the offspring can be separately populated by executing a more or less arbitrary “decoding” process on the content of G . Assuming the coreworld instruction set is Turing complete this decoding can, in principle, involve any arbitrary Turing computation.¹⁰ The P part of the agent is the only part that is directly executed during this process; and it will decompose straightforwardly into program sections having the functionality of the von Neumann components A , B and C ; any remaining part of P , which is functional and allows the agent to perform behaviours unrelated to reproduction, will correspond to von Neumann’s D .

Experimental investigation of the implementation and evolutionary behaviour of such agents is currently underway in both Tierra and Avida. For my immediate purposes here I simply mention some of the specific questions to be studied:

- How frequently are viable mutations of the genotype-phenotype map ($\psi()$) observed? How does this vary with the specific initial choice of map?
- Is the structure of the von Neumann architecture (the decomposition of M into $G + P$, and the further decomposition of P into $A + B + C + D$) stable under evolution? Or do the roles of these components become blurred?
- More particularly, can the reproduction architecture revert back to simple self-inspection?

¹⁰In the case of Tierra, Turing completeness was originally demonstrated by Maley (1994), albeit it turned out to be extremely clumsy due to the lack of instructions to directly move data between memory and CPU registers. A similar limitation affects the default configuration of Avida. However, in both cases it is straightforward to enable additional instructions to remove these limitations.

- To the extent that the reproduction architecture does *not* revert back to self-inspection, does the seeding with a von Neumann ancestor alter the typical evolutionary dynamics of these systems? For example, does the typical parasite (hyperparasite etc.) phenomena of Tierra still occur?

Minimal Biological Self-Reproduction

Having now articulated the two clearly distinct abstract approaches to self-reproduction — the von Neumann indirect, genotype-phenotype architecture, and the approach of comprehensive self-inspection and inspection — let us consider in more detail the relation between these and biological self-reproduction. I will focus on primitive, minimal, forms of biological self-reproduction on the basis that these give the best possibility of identifying analogues to these abstract mechanisms.

Von Neumann’s work was presumably grounded in some prior biological knowledge — including the conceptual distinction between genotype and phenotype, the known fact that the genotype had a material basis in the chromosome(s), and that it admitted of some degree of linear decomposition in the form of linkage maps among discrete, heritable, “genes”. He was likely also aware of Schrödinger’s specific suggestion that the chromosomes involved some kind of “aperiodic crystal” structure (Schrödinger, 1944). However, von Neumann still devised his abstract architecture around five years¹¹ before the detailed molecular structure of DNA was first identified (Watson and Crick, 1953). It was striking then that DNA turned out to be a polymer with a linear primary structure, where there are four distinct possible monomers that can be used at each position, in arbitrary sequence, while still being compatible with the overall structure. Further, during (cellular) reproduction, this sequence is precisely copied to a separate DNA molecule that is distributed to the offspring. Subsequent work in molecular biology demonstrated further that there exists active molecular machinery which can “decode” (“translate”) the information sequence from the DNA to produce all the key functional and structural components of the cell, primarily in the form of proteins. In particular, there is a very well defined (and almost universal) coding between DNA sequence and protein sequence, which is *prima facie* strongly reminiscent of the $\psi()$ mapping in the von Neumann architecture.

In more detail then, it is possible to identify specific molecular analogues of several elements of von Neumann’s architecture:

- G : Corresponds to DNA; in the case of bacteria, literally a single DNA chromosome.
- P : All the active molecular machinery, coded for by the DNA genome. Largely made up of structural and functional proteins. These can be partially refined as:

¹¹First publicly presented at the 1948 Hixon Symposium (von Neumann, 1951).

- *A*: The protein synthesis machinery, particularly the ribosomes, but also including the DNA to RNA transcription enzymes, and a number of other essential components.
- *B*: The DNA replication enzymes (DNA polymerase and others).
- *C*: DNA transcription regulators etc.
- *D*: All remaining protein (and RNA) components, not directly involved in reproduction.

While these analogies are striking, they should not be overstated either. There are also many points of contrast:

- DNA is double stranded, and this is an essential feature of the molecular mechanism for sequence replication (“inspection”).
- While bacteria do have just a single DNA molecule, this is circular rather than strictly linear. More complex organisms have a genome organised into multiple chromosomes; and may retain multiple versions of each (diploidy or polyploidy).
- Biological reproduction can and does commonly involve transfer of DNA between separate individuals (horizontal gene transfer, sexual recombination).
- The von Neumann self-reproduction cycle is largely sequential; whereas, while some cellular processes do have a somewhat sequential pattern (e.g., DNA replication and transcription, mRNA translation), most cellular processes proceed concurrently and asynchronously.
- While there is a clear and well established molecular mapping between DNA sequence and protein sequence, the latter alone certainly does not represent a complete specification of even phenotype (morphology, behaviour etc.) even at the bacterial level; so von Neumann’s abstract mapping $\psi()$ encompasses much more than just the so called “genetic code” — the latter is, at best, one initial step in this overall mapping.
- While proteins play a dominant functional role in the cell, RNA molecules also have some critical and pervasive functions. This is significant in terms of the architectural analogy in that RNAs are only transcribed from DNA — not *translated* from it. They form an intermediary in the translation to proteins (as messenger- or mRNAs) but also play active roles as enzymes (so-called ribozymes) including as important components in the ribosomes, and in the form of the transfer- or tRNAs which mediate the translation of mRNA codons (triplets of bases) into specific amino acids (protein monomers) according to the genetic code. There is no direct analog in the von Neumann architecture for the RNAs.

Prima facie the role of RNAs in the translation process might call into question whether even the genetic code component of the genotype-phenotype mapping is itself “encoded” into the genome in a self-referential manner; i.e., whether this arrangement satisfies the Pattee criterion of semantic closure. However, it turns out that the key determinants of the specific coding relationships are not RNAs but protein enzymes (the aminoacyl tRNA synthetases). These are, therefore, coded for in the genome according to the decoding they themselves specify, so this subtle aspect of the von Neumann architecture actually does hold reasonably well (cf., Hofstadter, 1985).

Finally, let us consider whether or to what extent the pure self-inspection mechanism of self-reproduction, traditionally employed in coreworld systems, has any direct biological analog. It is clear that no cellular organism has this self-inspection character. Viruses appear like a better candidate (and, indeed, this underlies the use of the term “computer virus” for malware which functions to reproduce itself “in the wild” of computer networks).

However, biological viruses rely on exploitation of molecular machinery in host cells for replication (copying) of the viral genome, and generally for transcription and/or translation to produce additional functional components (protein sheaths etc.) required for effective propagation. So biological viruses cannot reasonably be said to *self-reproduce* in the manner of traditional coreworld reproducers. It might be argued here that even coreworld reproducers need access to *some* external resources to function — but these are just the primitive, unstructured, “raw materials” of memory and CPUs provided by the coreworld universe. This is not comparable or analogous to the reliance of a biological virus on exploiting the already structured, complex, machinery of host cells.

A better analogy would appear to be to the *in vitro* molecular evolution systems, in which “naked” RNA molecules replicate by direct template inspection, and can give rise to perfectly Darwinian evolutionary dynamics (Joyce, 2007). Admittedly, this is no longer a “natural” biological system, but it is derived from real biological materials. This does compare well to the coreworld self-inspectors in one important respect: the single sequence of monomers in the molecules essentially functions simultaneously as both genome and phenome: the complete sequence is replicated (copied), and also directly determines (via the folded three-dimensional structure) the relevant enzymatic (ribozyme) activity. But the analogy is still weak in that such systems require the external provision not just of (already quite complex!) “raw materials” in the form of activated nucleotides, but also very complex protein enzymes to mediate the replication itself (e.g., $Q\beta$ -replicase).

The last analogy we might consider is with the molecular replicators of the so-called RNA-World hypothesis (Gilbert, 1986). In this case it is assumed that there must have ex-

isted at least some these RNAs with the ability to function as RNA-polymerases (ribozyme RNA-replicases). Since these would then be able to make copies of themselves, it would seem that these would be quite precisely analogous to coreworld self-inspectors. Admittedly, this would still be somewhat hypothetical: no such general purpose ribozyme RNA-replicases have been identified to date. But more importantly, even here, there is still one crucial *dis*-analogy. Even if ribozyme RNA-replicases are possible, it is not supposed that one individual molecular could literally function to replicate *its own* sequence. Rather, it is assumed that it could replicate the sequence of another already existing copy of itself. This may seem like a minor detail; but in fact the selectional dynamics of such systems would be very different indeed (showing hyperbolic rather than exponential growth, and a typical phenomenon not of Darwinian “survival of the fittest”, but “survival of the most common”). Consequently, the resulting evolutionary phenomena would also be expected to be very different indeed from anything normally occurring in coreworld systems.

Conclusion

This paper has set out to review and compare the two “canonical” forms of machine self-reproduction that have been proposed in the field of Artificial Life. Of these, the simpler, self-inspection, mechanism has been fully realised in the so-called coreworld systems, and the resulting evolutionary dynamics have been the subject of extensive experimental investigations. The other, the von Neumann architecture, has largely been studied in the context of cellular automaton universes. It has only recently been realised in any practical experimental system; and only in forms where, quite aside from very high computational demands, significant evolutionary dynamics are impossible even in principle due to the intrinsic fragility of the reproducing agents. It has been pointed out, nonetheless, that there are grounds for supposing that the von Neumann architecture may facilitate certain kinds of evolutionary exploration that are not possible at all in systems based exclusively on self-inspectors; and that moreover, while both these forms of self-reproduction are very abstract compared to any real biological reproducers (or even hypothetical ones such as the replicating molecules of the RNA-world), the von Neumann architecture does admit significantly more and deeper points of analogy with real molecular biology than self-inspectors do. An outline has been given of how von Neumann architecture reproducers could, in fact, be realised in coreworld systems; and some specific questions have been formulated for study in such a research program.

Acknowledgements

This work was supported by Complexity-NET Project EvoSym, via the Irish Research Council for Science, Engineering and Technology (IRCSET).

References

- Adami, C. (1998). *Artificial Life, An Introduction*. Springer.
- Adami, C. and Brown, C. T. (1994). Evolutionary learning in the 2D artificial life system “Avida”. In Brooks, R. A. and Maes, P., editors, *Proc. Artificial Life IV*, pages 377–381. MIT Press.
- Barricelli, N. (1957). Symbiogenetic evolution processes realized by artificial methods. *Methodos*, 9(35-36):143–182.
- Burks, A. W., editor (1966). *Theory of Self-Reproducing Automata [by] John von Neumann*. University of Illinois Press, Urbana.
- Gilbert, W. (1986). Origin of life: The RNA world. *Nature*, 319:618.
- Hofstadter, D. (1985). The genetic code: Arbitrary. In *Metamagical Themas: Questing for the Essence of Mind and Pattern*, page 671–699. Penguin Books, London.
- Joyce, G. F. (2007). Forty years of in vitro evolution. *Angewandte Chemie (International Ed. in English)*, 46(34):6420–6436. PMID: 17634987.
- Laing, R. A. (1977). Automaton models of reproduction by self-inspection. *Journal of Theoretical Biology*, 66:437–456.
- Maley, C. C. (1994). The computational completeness of Ray’s Terran assembly language. In Langton, C. G., editor, *Artificial life III*, pages 503–514. Addison-Wesley.
- McMullin, B. (2000). John von Neumann and the evolutionary growth of complexity: Looking backwards, looking forwards. *Artificial Life*, 6(4):347–361.
- Morita, K. and Imai, K. (1996). A simple self-reproducing cellular automaton with shape-encoding mechanism. In Langton, C. G. and Shimohara, K., editors, *Artificial Life V*, page 489–496. MIT Press, Nara, Japan.
- Pattee, H. (1982). Cell psychology: an evolutionary approach to the symbol-matter problem. *Cognition and Brain Theory*, 5(4):325–341.
- Ray, T. S. (1992). An approach to the synthesis of life. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II*, volume X, pages 371–408. Addison-Wesley Publishing Company, Inc., Redwood City, California.
- Ray, T. S. (1994). An evolutionary approach to synthetic biology: Zen and the art of creating life. *Artificial Life*, 1:179–209.
- Schrödinger, E. (1944). *What Is Life?* Cambridge University Press, 1st edition edition.
- von Neumann, J. (1951). The general and logical theory of automata. In Jeffress, L. A., editor, *Cerebral Mechanisms in Behavior—The Hixon Symposium*, pages 1–41. John Wiley and Sons.
- Watson, J. D. and Crick, F. H. C. (1953). Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, 171:737–738.
- Wikipedia contributors (2012). Von neumann universal constructor. Page Version ID: 487564813.

Evolutionary Dynamics and Ecosystems Feedback in Two Dimensional Daisyworld

Dharani Punithan and RI (Bob) McKay

Structural Complexity Laboratory,
Department of Computer Science and Engineering,
Seoul National University, South Korea.
{punithan.dharani,rimsnucse}@gmail.com

Abstract

We introduce replicator-mutator mechanisms from evolutionary dynamics into a two-dimensional daisyworld model, thereby coupling evolutionary changes with daisyworld's bi-directional feedback between biota and environment. Daisyworld continues to self-regulate in the presence of these evolutionary forces. The most interesting behaviours, exhibiting a complex and dynamic dance through space and time in species' abundance, emerges through the introduction of additive spatio-temporal random perturbations in the form of thermal noise. The balance between ecosystem feedback and fluctuations in the ecosystem determines the spatial coexistence of domains of dominance between daisy species and their mutants or adaptants.

Introduction

Evolutionary studies have highlighted the importance of biota-environment feedback in evolutionary dynamics, for example, niche construction (Odling-Smee et al., 2003), extended phenotypes (Dawkins, 1999). Biota-environment feedback is inherent in daisyworld models, so we have chosen to extend the basic daisyworld model with evolutionary dynamics based on the replicator-mutator equation (RME) (Hofbauer and Sigmund, 2003).

The three fundamental factors in Darwinian evolution are replication (entities reproducing themselves), mutation (producing small variations in transforming to a new entity) and selection (passing the fitter entities to later generations). These factors determine the population dynamics: changes in population size and evolution of new populations. Populations are the fundamental basis of evolution; individuals can change over time, but only populations evolve (Nowak, 2006). We focus our attention on population dynamics in daisyworld with evolutionary change.

Generally, selection arises as a consequence of competition, commonly due to prey-predator relationships or resource limits. In the daisyworld of Watson and Lovelock (1983), daisies compete for space and hence for light. Daisyworld also incorporates a feedback mechanism: different daisy species affect the temperature, and the temperature in turn affects daisy survival, and hence selection. Thus the

classic daisyworld realises the competition and natural selection of an evolutionary framework.

What is less studied in classical Darwinian evolution is the global feedback between biota and environment; conversely, the components that are omitted in the original daisyworld model are mutation and adaptation. In this field, it is common to distinguish between evolution of the daisies in ways which change their effect on the environment (in this case, albedo), which is referred to simply as mutation; and evolution of the daisies in ways which change their response to the environment (growth curve with temperature), generally referred to as adaptation. Thus a simple evolutionary daisyworld can incorporate all these factors, through 1) influence of temperature on daisies, 2) influence of daisies on temperature, 3) mutation of daisies and 4) adaptation of daisies to the temperature.

A number of researchers have studied evolution in daisyworld. Mutation was introduced into daisyworld by Lovelock (1992) and expanded by Lenton et al. (1998); Lenton and Lovelock (2001). Adaptation in daisyworld was studied by Lenton and Lovelock (2000). For a more detailed survey, please see Wood et al. (2008, section 4).

In this paper, we model the population dynamics of daisies using the replicator-mutator equation (RME) of evolutionary dynamics, in a diffusively coupled logistic lattice of daisyworld - a model which has not been studied previously. RME has the advantage of expressing replication, mutation and selection mechanisms within a single framework; these evolutionary changes can be influenced by external environment factors such as temperature or abundance of species. Also, the interaction with the environment affects the survival of daisies and changes their adaptive fitness. We have analysed the population dynamics of our daisyworld model through allowing a range of fluctuations in temperature, studying its effects on the evolutionary behaviour of the daisyworld. We did this by introducing ecosystem disturbance in the form of additive spatio-temporal Gaussian white noise (García-Ojalvo and Sancho, 1999). The scale of fluctuation is controlled by the noise level.

Background

Daisyworld Dynamics

In the daisyworld model of Watson and Lovelock (1983), there are two life forms (daisies), identical except for two aspects of their phenotype: one (black daisies) has low albedo and a preference for (i.e. faster growth in) low temperatures; the other (white daisies) has high albedo and a preference for high temperatures. Black daisies warm the planet by absorbing the solar heat; but naturally (since they themselves may get even hotter) prefer a cooler climate. White daisies cool the planet (more than bare ground) by reflecting heat, but prefer a warmer climate. The variation in the trait (petal colour or albedo – the environment-altering trait) affects their fitness by altering the temperature, which in turn determines the growth rate. Thus changes in fitness cause changes in the distribution of daisies, and changes in distribution of daisies cause changes in the fitness. This forms the fundamental feedback loop – a cycle which repeats indefinitely. Hence the temperature is heavily influenced by the abundance of black and white daisies, and competition for space between the two species self-regulates the planet. In this scenario, life does not merely influence the environment, but regulates it in a way that is suitable for itself.

Replicator-Mutator Dynamics

Replicator-mutator (aka selection-mutation) dynamics are embodied in the RME (equation 1):

$$\dot{x}_i = \sum_{j=1}^n x_j f_j q_{ji} - \phi x_i, i = 1, \dots, n \quad (1)$$

Here x_i is the population proportion of type i , n is the total species, $Q = [q_{ij}]$ is the mutation matrix, f_j defines the selection dynamics $f_j = f_0 + \sum_j a_{ji} x_j$ and $\phi = \sum_i x_i f_i$ is the average fitness. f_0 is the intrinsic fitness and $A = [a_{ij}]$ is a reward matrix.

In addition to biological evolution (Bürger, 1998), RME has been used to model evolution in language (Nowak et al., 2001), culture, behaviour in social networks (Olfati-Saber, 2007) and evolutionary game theory (Brenner, 1998).

Model

The ecosystem based on our daisyworld model is constructed on a diffusively coupled 2D toroidal regular lattice ($N \times N$) of locally chaotic oscillators; which describes population growth as well as population dispersal - a metapopulation lattice model. Each cell is viewed as a habitat, with a maximum carrying capacity of 10,000 individuals. The habitats are randomly initialised with a population size in $[0, 100]$ for both species. The temperature is initialised to 295.5K. The diffusion of species and temperature is determined by the neighbourhood model – von Neumann neigh-

bourhoods, consisting of a central cell and its four orthogonal neighbours. We use Laplacian diffusion for both species and temperature diffusion. The model parameters are represented in Table 1.

Table 1: Daisyworld Parameter Settings

Parameter	Value
Heat Capacity (C) $Wm^{-2}K^{-1}$	2500
Diffusion constant (D_T) $Wm^{-2}K^{-1}$	500
Stefan-Boltzmann constant (σ_B) $E^{-8}Wm^{-2}K^{-4}$	5.67
Luminosity (L)	1
Solar Insolation (S) Wm^{-2}	864.65
Dispersion rate of daisies (D)	0.2
Noise Level	$[0, 3]$

The dynamics of the n species is governed by the following set of equations.

Albedo: The albedo (A) at a particular habitat is computed as the weighted average in equation 2:

$$A = A_{ground}(1 - \sum_i^n \alpha_i) + \sum_i^n A_i \alpha_i \quad (2)$$

In this equation, n is number of daisy types, A_i is the albedo of each daisy type and α_i is the corresponding proportion of daisy cover. In our experiments, bare ground is completely occupied within the first few epochs.

Temperature: The local temperature is governed by diffusion, heat radiation, solar absorption and Gaussian noise:

$$C \frac{dT(l)}{dt} = D_T \nabla^2 T(l) - \sigma_B T(l)^4 + SL(1 - A(l)) + C\epsilon(l) \quad (3)$$

$C = 2500$ is the heat capacity, T is the temperature, l is the spatial location, $D_T = 500$ is the diffusion constant, $\nabla^2 T$ is the Laplacian operator, σ_B is the Stefan-Boltzmann constant, S is the solar constant, L is the luminosity, A is the albedo (environment altering trait) and ϵ represents Gaussian white noise (white in space and time with mean zero and standard deviation 1.0) multiplied by the noise level. T , A , ϵ vary with location l .

Patch Temperature: Daisies at a location are divided (by species) into patches, whose temperature may vary from the local temperature. The albedo of the daisies in the patch determines the patch temperature:

$$T_i = q(A - A_i) + T \quad (4)$$

where T_i is the patch temperature, $q = 20$ is a constant; refer (Lovelock, 1992), A is the albedo of a habitat, A_i is the albedo of each daisy type and T is the local temperature of a habitat from equation 3.

Growth: The growth curve of daisies is as an inverted parabola defined in equation 5:

$$\beta(T) = \max(0, m(1 - [\frac{(T_{opt} - T)}{r}]^2)) \quad (5)$$

T is the local temperature, m is the peak growth rate, T_{opt} is the optimal temperature of the species and r denotes the range of temperature tolerance. In our model, $m = 1$ and $r = 17.5$.

Density Dependent Fitness: We use absolute numbers for population size (P) rather than the relative frequencies of the classical form (equation 1). To prevent unrealistic exponential growth or decay, we formulated our fitness (f) based on a carrying capacity (K) such that $f < 0$ for $P > K$ and $f > 0$ for $P < K$. The death rate of each daisy type is dependent on the total density of daisies. The corresponding fitness function is (equation 6):

$$f = \beta * (1 - \frac{P}{K}) - \gamma * (\frac{P}{K}) \quad (6)$$

β is the growth rate from equation 5, $\gamma = 0.3$ is the death rate, P is total population size and K the carrying capacity.

Population Size: The local population abundance is based on replicator-mutator dynamics. We start from a variant of the discrete form (Page and Nowak, 2002) of RME (equation 1) shown in equation 7:

$$P_{i(l)} = \frac{\sum_{j=1}^n P_{j(l)} f_{j(l)} q_{ji}}{\phi}, i = 1, \dots, n \quad (7)$$

where P_i is the proportion of the population of type i , l represents a spatial location, n is the number of phenotypes, and f_j defines the selection dynamics. Rather than separate reward (payoff) and mutation matrices, we use a combined replicator-mutator matrix $Q = [q_{ij}]$ – a doubly stochastic matrix – defining the evolutionary properties and satisfying $\sum_i q_{ij} = \sum_j q_{ij} = 1$. The diagonal of the matrix defines the replication rate, while other entries defines the mutation rates. $\phi = \sum_i P_i f_i$ is the average fitness, and is used as a normalisation term to ensure the population doesn't exceed its bounds.

Since our fitness function is density dependent and directly ensures the orbits are bounded in the region $P \leq K$, the term ϕ is not needed. However we need to incorporate diffusion of individuals. Combining this, we can define the change in population as in equation 8:

$$P_{i(l)} = D + \sum_{i=1}^n P_{i(l)} f_{i(l)} q_{ii} + \sum_{j \neq i}^n P_{j(l)} f_{j(l)} q_{ji} \quad (8)$$

D is the Laplace-diffused population. Since we don't use a reward matrix, the discrete time fitness (selection coefficient) is just the intrinsic fitness, leading to equation 6.

The fitness of a particular type of daisies changes, depending on the abundance of both itself and the other species because of the life-environment feedback. Here, f affects P and depends on β which depends on T which depends on A which in turn depends on P , thus forming a tightly coupled system with an indefinite feedback loop. The model incorporates two types of feedback: from the life-environment effect of daisyworld and from the density-dependent nature of population dynamics. This model covers all the main ingredients of evolutionary population dynamics – reproduction, mutation, selection and spatial dispersion – with a bi-directional life-environment feedback effect.

Importance of Noise

Modelling nonlinear dynamics in a noisy world is very common in population ecology. The two key factors in studying population dynamics are the internal feedback which inhibits the exploding population and the external environmental (abiotic) variability which determines the population fluctuations (Begon et al., 1996). In our model, the first factor (internal feedback) results from the imposition of a carrying capacity and the second (external variability) from Gaussian noise. These external perturbations can cause the system to deviate from equilibrium. The noise level controls the scale fluctuations in the environment. Noise helps to introduce sufficient nonlinearity to observe complex dynamics such as limit cycles, quasi-periodicity and chaos (nonlinear attractors). Diffusion of temperature and dispersion of daisies also allow the system to break symmetry and generate interesting patterns in daisyworld (Punithan and McKay, 2012). In this paper, we focus on the evolutionary population dynamics of daisyworld, so we study the impact of the external variability in detail. In the absence of noise, the dynamics of the world is stationary (static attractor). But with the imposition of noise, we observe evolving patterns of life – so long as there is not too much. With higher levels, noise completely dominates the feedback, and the behaviour becomes random and uninteresting.

Results

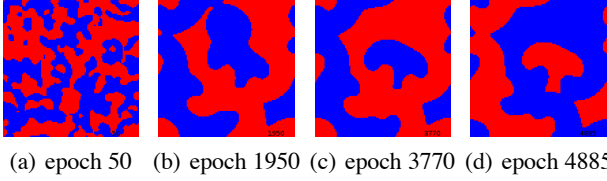
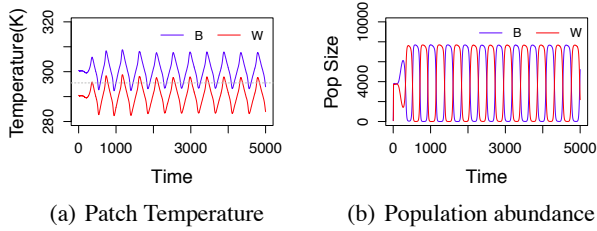
We analyse the changes in dominance by specific phenotypes (petal colour) under the effect of environmental perturbations (temperature). In our experiments, the optimal temperature and albedo of the original black, white daisies, their mutants and adaptants are tabulated in the Table 2; please refer Lenton and Lovelock (2001). In all analyses, we first examine the spatio-temporal population patterns (using snapshots of the system status over the 5000 epochs). We focus on the local dominance of species. We plot the local (temporal) behaviour at a single habitat, and the global dynamics (average temporal behaviour of the whole ecosystem).

Table 2: Optimal Temp. and albedo of original daisies and their mutants

	$T_{opt}(K)$	Albedo	Colours in Figures
Black	295.5	0.25	blue
White	295.5	0.75	red
Grey	295.5	0.5	green
Warm Black	300.5	0.25	cyan
Cool White	290.5	0.75	magenta

I. Baseline Scenario: No Mutation

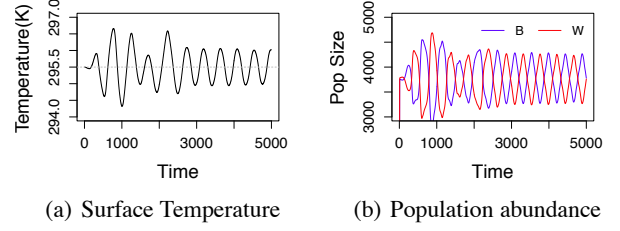
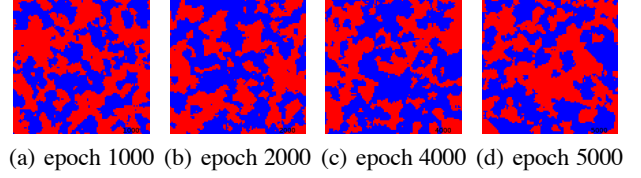
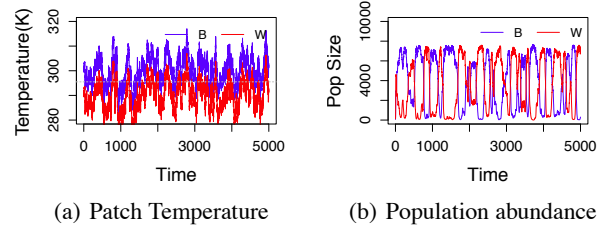
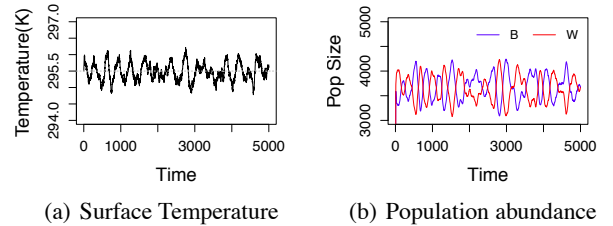
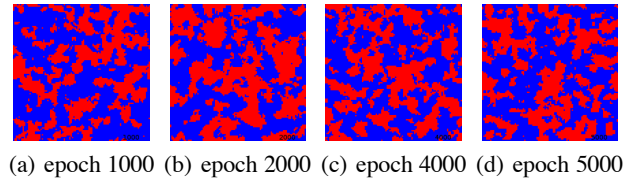
In this subsection, we present the behaviour of the system with no evolutionary change, as a baseline for comparisons. Since the level of noise is an important control in determining the effect of evolutionary change, we present results for different levels of noise.


 Figure 1: Population abundance without mutation, $D = 0.2$, $NL = 0.05$

 Figure 2: Local dynamics at habitat (55, 41) without mutation and $NL = 0.05$

Noise Level = 0.05 With low noise levels ($NL = 0.05$), periodic behaviour emerges from the system, see figure 1. The cycles in the local dynamics (figure 2) for both patch temperature and population confirm this. At the global level, the temperature (figure 3) is regulated around $295.5K$.

Noise Level = 3 With increased noise, rough turing like structures form – figure 4, and the periodic behaviour becomes less regular – figures 5 and 6.

Noise Level = 5 With further increase in noise, rough clusters ceaselessly form and dissolve (figure 7), and the periodic behaviour almost entirely disappears as in figures 8


 Figure 3: Global dynamics without mutation, $NL = 0.05$

 Figure 4: Population abundance without mutation, $D = 0.2$, $NL = 3$

 Figure 5: Local dynamics at habitat (55, 50) without mutation, $NL = 3$

 Figure 6: Global dynamics without mutation, $NL = 3$

 Figure 7: Population abundance without mutation, $D = 0.2$, $NL = 5$

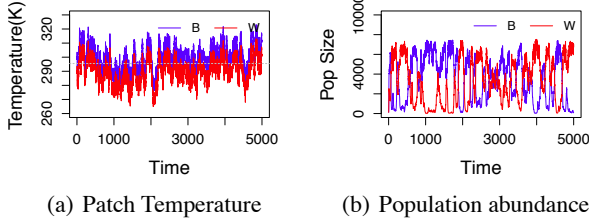


Figure 8: Local dynamics at habitat (56, 50) without mutation, $NL = 5$

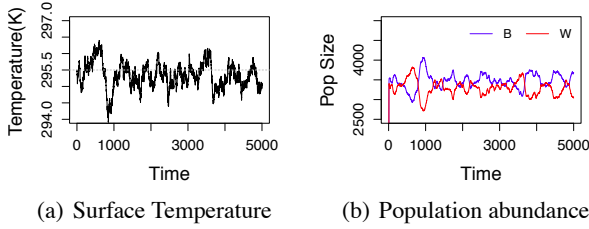


Figure 9: Global dynamics without mutation, $NL = 5$

and 9. We have reached the less interesting region in which noise dominates the environmental feedback; behaviours in this region are little affected by other parameters, so we omit it from consideration in the rest of the paper.

II. Phenotypic Variability – Mutation

The ability of mutations to generate new phenotypes is one of the hallmarks of Darwinian evolution. The black and white daisies in the model undergo random mutation, introducing a new phenotypic state – grey daisies ($T_{opt} = 295.5$ and albedo = 0.5). The evolutionary relationships between the species are defined in the replicator-mutator matrix (μ is the rate of random mutation and B, W, G denote Black, White and Grey daisies):

$$Q = \begin{matrix} & \begin{matrix} B & W & G \end{matrix} \\ \begin{matrix} B \\ W \\ G \end{matrix} & \begin{pmatrix} (1 - 0.1\mu) & 0 & 0.1\mu \\ 0 & (1 - 0.1\mu) & 0.1\mu \\ 0.1\mu & 0.1\mu & (1 - 2(0.1\mu)) \end{pmatrix} \end{matrix}$$

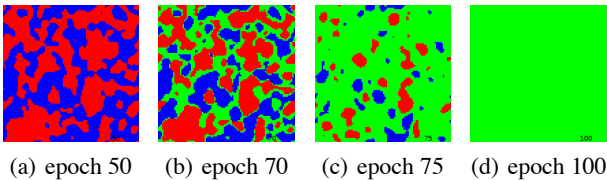


Figure 10: Population abundance with Grey daisies, $D = 0.2$, $NL = 0.05$

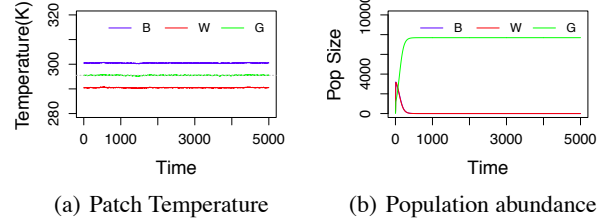


Figure 11: Local dynamics at habitat (57, 49) with Grey and $NL = 0.05$

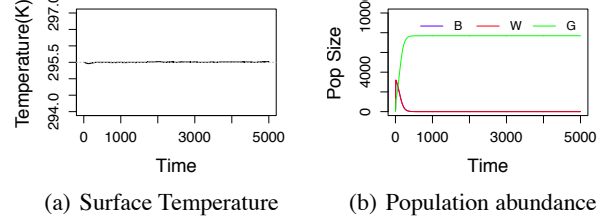


Figure 12: Global dynamics with Grey, $NL = 0.05$

Noise Level = 0.05 Grey daisies quickly dominate (figure 10) when the noise level is very low ($NL = 0.05$). Both local (figure 11(b)) and global (figure 12(b)) populations of the original black and white daisies drop very low, almost disappearing. This happens because the patch temperatures of black and white daisies are almost constant, above and below $295.5K$, so the growth of grey daisies is favoured: refer to figure 11(a). Since there is not much fluctuation in the local temperature due to the low noise level, grey daisies dominate the whole space. The temperature is regulated to $295.5K$ – see figure 12(a).

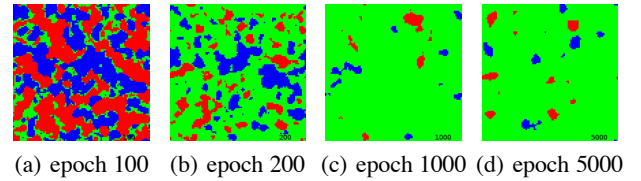


Figure 13: Population abundance with Grey daisies, $D = 0.2$, $NL = 2$

Noise Level = 2 When the noise level increases to 2, although the world is still rapidly dominated by grey daisies, the original black and white species can form distinct groups and dominate smaller regions (figure 13). All three species coexist locally as well as globally, mainly to due the patch temperatures. Black patches mostly fluctuate above $295.5K$, and white mostly below, with only grey fluctuating around the favoured $295.5K$ (figure 14(a)). The global

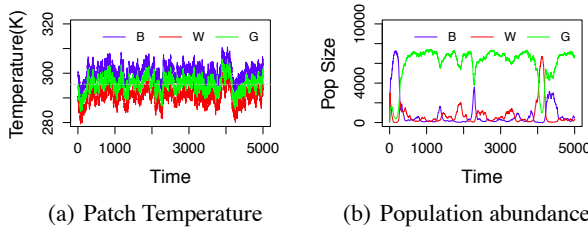


Figure 14: Local dynamics at habitat (54, 43) with Grey and $NL = 2$

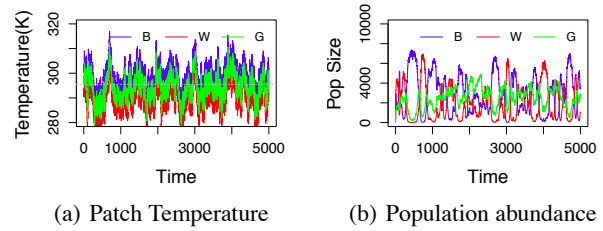


Figure 17: Local dynamics at habitat (51, 53) with Grey and $NL = 3$

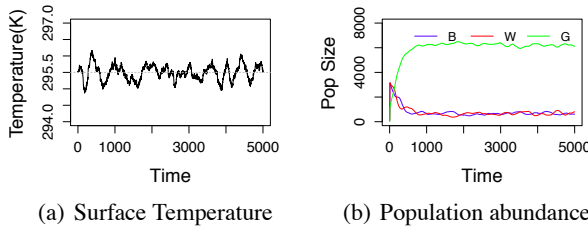


Figure 15: Global dynamics with Grey, $NL = 2$

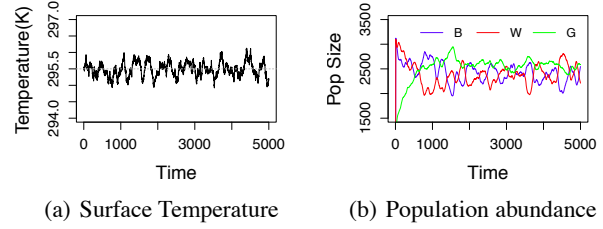
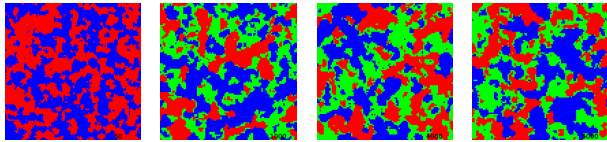


Figure 18: Global dynamics with Grey, $NL = 3$

temperature is regulated around $295.5K$ (figure 15(a)) with coexisting populations (figures 14(b) and 15(b)) but a majority of grey.



(a) epoch 50 (b) epoch 1000 (c) epoch 4000 (d) epoch 5000

Figure 16: Population abundance with Grey daisies, $D = 0.2$, $NL = 3$

Noise Level = 3 Further increasing the noise level to 3 results in all species dominating local regions, with highly evolving and dynamic (figure 16) behaviours. The local patch temperatures fluctuate widely around the mean $295.5K$ (figure 17(a)). The global temperature is regulated to around $295.5K$ (figure 18(a)) with all species coexisting locally (figure 17(b)) and globally (figure 18(b)).

To summarise, the grey mutant form dominates the whole ecosystem when there are small temperature fluctuations ($NL = 0.05$). As the noise level increases to ($NL = 2$), the black and white daisies form distinct population groups. Further increase in noise level ($NL = 3$), allow all three species to dominate locally, and they coexist both locally and globally. Thus we see the determination of evolutionary dominance through environmental influences.

III. Mutation without Life-Environment Feedback

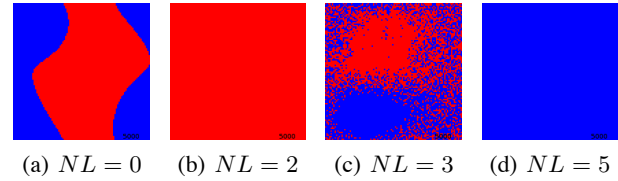


Figure 19: Population abundance without feedback, $D = 0.2$ at epoch 5000

We observed population spatial structures with mutation but excluding life-environment feedback (i.e. temperature is not influenced by the daisies at all) for different noise levels (figure 19). When we compare these snapshots (no grey dominance) with figures 10 and 16, we can see that the balance between feedback and noise plays a vital role in the coexistence of dominance of the original daisies and their mutants. The dynamics we see, for example, in figure 16 is due to the interplay of noise and feedback, rather than either alone.

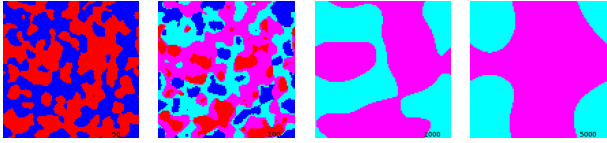
IV. Adaptation – Exploring New Environments

In the absence of other species, black daisies will generate a very hot environment unsuited to them. Instead of mutating to reduce their environmental effect, they can also adapt to tolerate these higher temperatures, generating “warm black daisies”. Similarly, white daisies may adapt to a cold environment, giving “cool white daisies”. We explore the effects

of warm black ($T_{opt} = 300.5$, $albedo = 0.25$) and cool white daisies ($T_{opt} = 290.5$, $albedo = 0.75$). In all patch temperature plots, the patch temperature trajectories of black and warm black daisies superimpose and similarly for white and cool white – inevitably, because the originals and their adaptants have the same albedo.

The evolutionary properties are by the following replicator-adaptor matrix (WB and CW stand for warm black and cool white daisies):

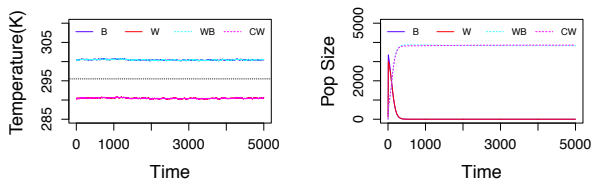
$$Q = \begin{matrix} & \begin{matrix} B & W & WB & CW \end{matrix} \\ \begin{matrix} B \\ W \\ WB \\ CW \end{matrix} & \begin{pmatrix} (1-0.1\mu) & 0 & 0.1\mu & 0 \\ 0 & (1-0.1\mu) & 0 & 0.1\mu \\ 0.1\mu & 0 & (1-0.1\mu) & 0 \\ 0 & 0.1\mu & 0 & (1-0.1\mu) \end{pmatrix} \end{matrix}$$



(a) epoch 50 (b) epoch 100 (c) epoch 1000 (d) epoch 5000

Figure 20: Population abundance with Warm Black and Cool White daisies, $D = 0.2$, $NL = 0.05$

Noise Level = 0.05 When the noise level is low (0.05), the original daisies almost disappear (figures 21(b) and 22(b)) and the warm black and cool white self-organise, forming patterns with huge clusters that dominate the ecosystem (figure 20). The global surface temperature is regulated to 295.5K (figure 22(a)). Though the patch temperatures of black and warm black are identical (also for white and cool white) (figure 21(a)), the adaptants are superior to the existing phenotypes, and rapidly dominate.



(a) Patch Temperature (b) Population abundance

Figure 21: Local dynamics at habitat (51, 50) with Warm Black and Cool White, $NL = 0.05$

Noise Level = 2 At a noise level of 2, both warm black and cool white, self-organise to form Turing-like patterns, while the originals, black and white, dominate in smaller regions here and there (figure 23). The local and global dynamics are shown in figures 24 and 25.

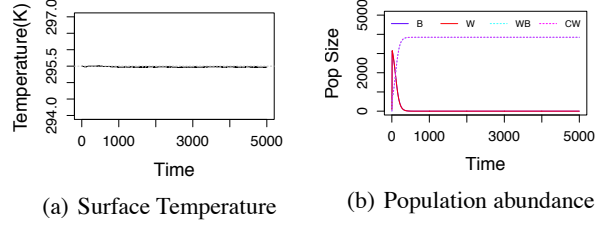
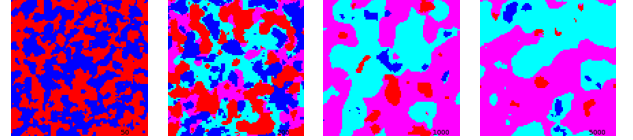
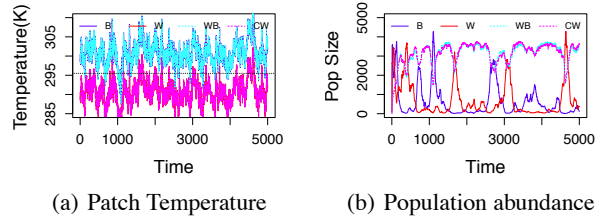


Figure 22: Global dynamics with Warm Black and Cool White, $NL = 0.05$



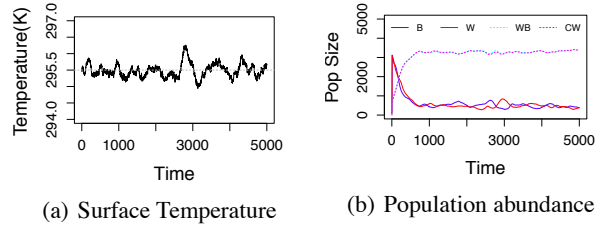
(a) epoch 50 (b) epoch 200 (c) epoch 1000 (d) epoch 5000

Figure 23: Population abundance with Warm Black and Cool White daisies, $D = 0.2$, $NL = 2$



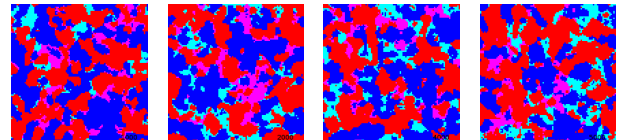
(a) Patch Temperature (b) Population abundance

Figure 24: Local dynamics at habitat (57, 55) with Warm Black and Cool White, $NL = 2$



(a) Surface Temperature (b) Population abundance

Figure 25: Global dynamics with Warm Black and Cool White, $NL = 2$



(a) epoch 1000 (b) epoch 2000 (c) epoch 4000 (d) epoch 5000

Figure 26: Population abundance with Warm Black and Cool White daisies, $D = 0.2$, $NL = 3$

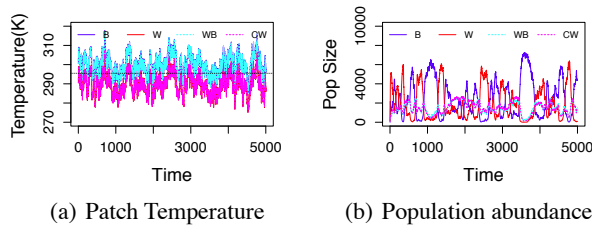


Figure 27: Local dynamics at habitat (52, 61) with Warm Black and Cool White, $NL = 3$

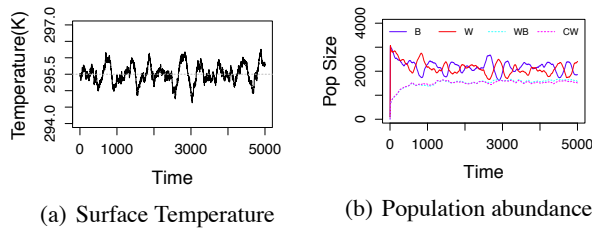


Figure 28: Global dynamics with Warm Black and Cool White, $NL = 3$

Noise Level = 3 All four phenotypes coexist, each dominating the ecosystem locally (figure 26). The overall behaviour is revealed in figures 27 and 28.

Conclusion

The evolutionary daisyworld model – replication with mutation and adaptation – presented in this paper illustrates global surface temperature regulation around $295.5K$ as in the original models. Thus demonstrating its robustness in homeostatic self-regulation in scenarios such as evolution of the environment altering trait (albedo) and adaptive evolution of optimal temperature. Temporal fluctuation in temperature, due to ecosystem disturbance, introduces nonlinearity into the daisyworld, leading to the most interesting behaviours. With very low noise, we observe monotonous life (quiescent daisy dominance states). Without feedback or with very high noise, we observe suppression of mutants. Thus, these results underline the importance of balance between ecosystem feedback and ecosystem disturbance in generating spatially coexistence of domains of dominance among the original daisies and their mutants.

Acknowledgements

This research was supported by the Basic Science Research Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (Project No. 2012-004841), and the BK21-IT program of MEST. The ICT at Seoul National University

provided research facilities for the study. We thank Ilun Science and Technology Foundation for their generous support of Dharani Punithan.

References

- Begon, M., Harper, J. L., and Townsend, C. R. (1996). *Ecology: individuals, populations, and communities*. Blackwell.
- Brenner, T. (1998). Can evolutionary algorithms describe learning processes? *Journal of Evolutionary Economics*, 8:271–283.
- Bürger, R. (1998). Mathematical properties of mutation-selection models. *Genetica*, 102-103:279–298.
- Dawkins, R. (1999). *The extended phenotype: The long reach of the gene*. Oxford University Press, USA.
- García-Ojalvo, J. and Sancho, J. M. (1999). *Noise in spatially extended systems*. Springer.
- Hofbauer, J. and Sigmund, K. (2003). Evolutionary game dynamics. *Bulletin of the American Mathematical Society*, 40(4):479 – 519.
- Lenton, T. M. et al. (1998). Gaia and natural selection. *Nature*, 394(6692):439–447.
- Lenton, T. M. and Lovelock, J. E. (2000). Daisyworld is darwinian: constraints on adaptation are important for planetary self-regulation. *Journal of Theoretical Biology*, 206(1):109–114.
- Lenton, T. M. and Lovelock, J. E. (2001). Daisyworld revisited: quantifying biological effects on planetary self-regulation. *Tellus B*, 53(3):288–305.
- Lovelock, J. E. (1992). A numerical model for biodiversity. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 338(1286):383–391.
- Nowak, M. A. (2006). *Evolutionary dynamics: exploring the equations of life*. Belknap Press.
- Nowak, M. A., Komarova, N. L., and Niyogi, P. (2001). Evolution of universal grammar. *Science*, 291(5501):114.
- Odling-Smee, F. J., Laland, K. N., and Feldman, M. W. (2003). *Niche Construction: the Neglected Process in Evolution*, volume 37 of *Monographs in Population Biology*. Princeton Univ. Press.
- Olfati-Saber, R. (2007). Evolutionary dynamics of behavior in social networks. In *Decision and Control, 2007 46th IEEE Conference on*, pages 4051–4056. Ieee.
- Page, K. M. and Nowak, M. A. (2002). Unifying evolutionary dynamics. *Journal of theoretical biology*, 219(1):93–98.
- Punithan, D. and McKay, R. (2012). Self-organizing spatio-temporal pattern formation in two-dimensional daisyworld. *Self-Organizing Systems*, pages 72–83.
- Watson, A. J. and Lovelock, J. E. (1983). Biological homeostasis of the global environment: The parable of daisyworld. *Tellus B*, 35(4):284–289.
- Wood, A. J., Ackland, G. J., Dyke, J. G., Williams, H. T. P., and Lenton, T. M. (2008). Daisyworld: a review. *Reviews of Geophysics*, 46(1).

Adaptation and Divergence during Experimental Evolution of Multicellular *Saccharomyces cerevisiae*

Maria Rebolleda-Gomez^{1,2}, William Ratcliff^{1,2} and Michael Travisano^{1,2}

¹Ecology, Evolution and Behavior, 100 Ecology, 1987 Upper Buford Circle, University of Minnesota, St. Paul, MN 55108

²BioTechnology, 140 Gortner Lab, 1479 Gortner Ave, University of Minnesota, St. Paul, MN 55108-6106

rebol004@umn.edu

Abstract

The evolution of multicellularity was one of the key innovations in the history of life on Earth. Virtually all morphological and ecological diversity in macro-organisms builds upon the evolutionary potential associated with multicellularity. We examined the potential for ecological diversity to rapidly arise following transitions to multicellularity. Replicate microcosms containing the yeast *Saccharomyces cerevisiae* were maintained under serial transfer. Prior to transfer to fresh media each day, *S. cerevisiae* underwent settling selection via mild centrifugation. Those individuals reaching the bottom of the centrifuge tube were transferred to fresh media. After sixty days, all microcosms contained multicellular individuals that develop via mother-daughter adhesion. In nine of the ten microcosms, at least two distinctive morphological genotypes were evident at sixty days, and in eight of them, the variants were multicellular. We observed substantial morphological variation across replicates, with relatively little parallelism in the size of multicellular individuals or in the size variation within microcosms. These results suggest surprising amounts of contingency in the evolution of ecological diversity, and that “replaying life’s tape” would lead to divergent outcomes.

Introduction

The diversity of life is amazing, and its uneven distribution across taxa is a major puzzle in evolution. Why do some groups rapidly diversify and give rise to many different evolutionary units, whereas others diversify only into a few lineages? It has been proposed that the capacity of a lineage to diversify depends to a great extent on its capacity to re-invent itself (Crepet and Niklas, 2009). It depends on the evolutionary origin of innovations that increase the possible variation available for evolution (Maynard-Smith and Szathmáry, 1995; Sterenly, 2011). The origin of multicellularity—as an evolutionary innovation—increased the evolutionary potential of plants and animals by increasing the number of possible phenotypes and accessibility to further innovations that require the organization of multiple cells, like the tetrapod limb in animals or the flower in angiosperms. Other multicellular groups, like the volvocine algae, have only a few multicellular forms and remain far less diverse than plants, and their algal ancestors, the charophytes (Nedelcu and Michod, 2004).

It has been argued that major evolutionary changes, involving the reorganization of the organism as a whole, are largely contingent on their prior history. In this respect S. J. Gould (1989) said that as a succession of improbable events, “re-playing life’s tape” would lead to divergent outcomes. The role of historical contingency in evolutionary pathways, however, is still contentious, even with respect to the evolutionary origin of innovations (Travisano et al., 1995; Vermeij, 2005; Blount et al., 2008).

Mechanisms promoting diversity act at different scales (Whittaker 1960; Whittaker et al., 2001). The prevalence of historical factors in shaping life’s diversity might depend on the scale considered. Local diversity (α diversity), for example, has been mainly attributed to ecological dynamics of interactions between and within lineages; as well as the role of the environment over these interactions. Instead, turnover on a regional scale (β diversity), has been explained by historical and large-scale environmental changes like latitudinal temperature gradients (Whittaker et al., 2001). β diversity has been argued to be largely determined by the way in which ecological interactions and environmental factors have affected the composition of each community over time. In this sense, lineages’ turnover is very contingent on previous conditions.

Little consideration has been given to the problem of scale in the context of evolutionary innovations and diversification. How do major phenotypic changes affect diversity at a local scale? What is their impact on lineage turnover at a larger scale? How do evolutionary innovations interplay with the causes of diversity at these two levels?

Answering these questions has been experimentally challenging. Innovations like multicellularity evolved independently in separate lineages deep in the past (Bonner, 1998), and their implications for the subsequent evolution of diversity are difficult—to say the least—to infer from the fossil record. The ecological factors promoting diversity at the local scale are sometimes difficult to measure, and determining the proper scales for measurement is problematic due to continuity among different communities (Graham and Fine 2008; Fraser et al., 2009).

Microbes provide a good model to investigate ecological and evolutionary questions. These organisms have short generation times and are easily propagated in controlled environments. All these properties allow for high replicability and thus, high comparative power (Travisano 2009). Recently,

Ratcliff, et al., (2012) performed a selection experiment to study the evolution of multicellularity. Ten replicate populations were established by inoculation with a single strain of the yeast *Saccharomyces cerevisiae*. Every 24 hours these populations underwent settling selection via mild centrifugation. Those individuals reaching the bottom of the centrifuge tube were transferred to fresh media. After sixty days, all microcosms contained yeast multicellular individuals that develop via mother-daughter adhesion. This experiment provides a good system to evaluate the effect of the evolution of multicellularity on phenotypic diversity at two levels: within and across populations.

Methods

Strains and media

The strains used for this research were ten single genotypes isolated (single colony selection, repeated three times serially) from each of the 10 replicate populations of the original experiment (Ratcliff et al., 2012) after 60 transfers.. All strains were grown in liquid YPD (per liter: 10g yeast extract, 20g peptone, 20g dextrose). Colony isolation was performed after growth in YPD agar plates (15% agar). Six isolates from population one were used for the growth curves (big - 1, 2 and 5 and small - 3, 6 and 8).

Cluster size

Yeast was grown for 24 hrs at 30°C in 25 x 150 mm tubes with 10 mL of fresh YPD media shaken at 250 rpm. For conditioning, 100 µl of the culture were transferred to 10 ml of fresh media. These cultures were grown for other 24 hrs under the same conditions.

After 24 hours of growth, 100 µl was obtained from each culture and diluted 1:10 in 0.85% saline solution, from which 10 µl were placed in a hemocytometer chamber. Ten fields of view were photographed. Using the 4X objective and brightfield illumination, pictures were captured on an Olympus IX-70 with a Scion CFW-1310C camera. The acquisition properties were kept consistent throughout the experiment. Once captured, we removed the background of all images using a constant threshold value. We then measured the area of all clusters in each picture. All these image analyses were performed using ImageJ (NIH). To measure within-populations variation in cluster sizes we used captured images from two replicates for each isolate.

Growth rate

After 24 hrs of conditioning growth (at 30°C in 25 mm tubes with 10 ml of fresh YPD media shaken at 250 rpm), we transfer yeast to fresh media and estimated growth curves. 100 µl of liquid culture was diluted 1:100 into 10 ml of fresh YPD. The number of individuals was determined after zero, four and eight hours of growth by direct counting over the 10 different fields of view. Samples were diluted in 0.85% saline solution at a 1:10 dilution before counting. Three replicate tubes were inoculated with each isolate, and growth was measured for each replicate.

Data analysis

Individual size. Due to the presence of both “adult” clusters and juvenile offspring, the size distribution for each isolate was bimodal. In addition, some isolates had a higher proportion of offspring than full-grown clusters. To avoid including juveniles in assessing adult size, an arbitrary size threshold was established, and all the cluster areas below that value were eliminated from the analysis. We defined this threshold in such a way that almost all juvenile offspring were eliminated from the analysis, but none of the full-grown clusters (not even the smallest ones). Because of their great divergence, different threshold values were used for populations 9 and 10.

A nested REML ANOVA (isolate within replicate population, individual observations within isolate) was used to assess individual size and variation among and within replicate populations. The area of individual clusters was square-root transformed to normalize the data. Differences among isolates were then identified using a Tukey HSD test. Comparisons between isolates were performed with JMP Pro 9.0.2 (SAS Institute Inc., 2010). Estimates of α and β diversity were determined by calculating the square root of the genetic variation for individual size within replicates (α diversity) and across replicates (β diversity). Point estimates and 95% confidence intervals were determined by REML calculations.

Growth rate. To calculate the late growth rate (i.e. from 4 to 8 hrs; m_{8-4}) we fixed the y-intercept to the initial cell density and we calculated the growth rate over the entire eight hours and for the first four hours (see Lenski et al., 1991; Travisano 1996). The growth rate over the later four hours was computed by subtracting the rate from the entire eight hours’ growth rate and dividing it by the four hours of difference:

$$m_{8-4} = \frac{m_8 - m_4}{8 - 4}$$

The difference between earlier and late growth rates for the different isolates was evaluated with a two-way ANOVA.

Results and Discussion

Ten strains were isolated from each replicate population after 60 days of settling selection under uniform conditions. Multicellularity evolved in all ten replicate populations over the course of selection. The size of multicellular individuals was determined for each isolate and an unexpectedly high amount of diversity was found when comparing both different populations and isolates within each population (Figures 1-3). Out of the ten replicate populations, six different size classes could be distinguished (Figure 2) and there are clear size differences between cluster areas of small and big populations (Figure 1).

In prior microbial selection experiments, there is a considerable amount of convergence in adaptive traits because the environment is kept constant across populations (Lenski and Travisano 1994). Complex innovations sometimes require multiple steps and thus, are more likely to be historically

contingent. There are, however, experimental examples of convergence in complex novelties, despite contingency (Meyer et al. 2012). Overall, microbial evolution experiments have shown that diversification depends largely on ecological opportunities (Rainey and Travisano, 1998) and ecological interactions between different genotypes (Rainey and Travisano, 1998; Kerr et al., 2002; Friesen et al., 2004). Co-evolution and ecological dynamics might even determine the evolutionary outcomes. In this sense, evolutionary change might be contingent not only on previous changes but in the resulting ecological dynamics (Meyer et al. 2012). The evolution of multicellularity probably modifies both the environmental and the adaptive landscapes, potentially allowing for increased diversification (Ispolatov et al., 2011).

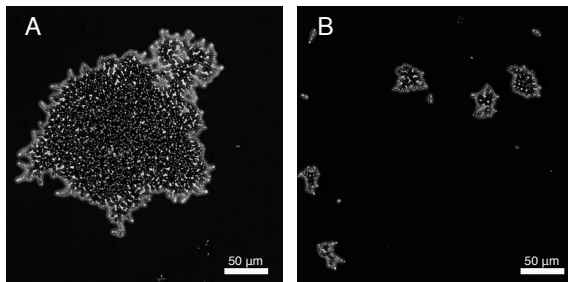


Figure 1. Divergence in individual size rapidly evolved during selection for settling. Shown are large individuals from population 3 (A) and small individuals from population 9 (B).

Diversity was found at two levels: across all replicate populations and within each population. Causes of diversity are probably different, depending upon the scale examined (i.e. across or within populations). Isolates within a population coexist and compete for the same resources, whereas genotypes from different populations are isolated from each other and therefore do not compete, but instead, have independent evolutionary histories. Thus, to further understand the relationship between the evolution of multicellularity and increased diversification, we need to distinguish both components, diversity within (α diversity) and between populations (β diversity).

The majority of the overall diversity is explained by differences across populations (high β diversity), whereas only a relative small percentage corresponds to α diversity (Figure 3). Patterns of β diversity are often explained by (i) differences in environmental conditions like temperature or precipitation (Quian et al., 2009); (ii) historical factors, like geographical isolation causing independent evolutionary histories (Qian et al., 2005), different patterns of migration (Quian, 2009) and stochastic processes involved in community assembly (Chase, 2010); (iii) or an interplay of ecological and historical factors (Grenner et al., 2004). These populations were all started with a single unicellular genotype propagated under uniform conditions across all ten populations. It is very likely that the observed β diversity is associated with the evolution of multicellularity and the evolutionary history of each population.

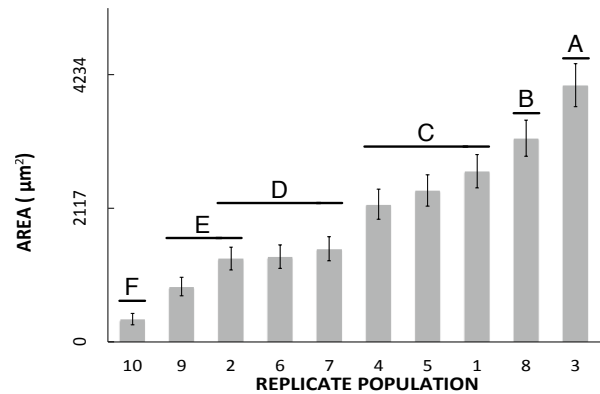


Figure 2. Replicate populations diverged in mean individual size during settling selection, from small (replicate 10) to large (replicate 3). Values shown are replicate means and jointly determined 95% CI. At least six different size classes could be statistically distinguished across all ten replicates (A – F), based on a Tukey HSD test with a significance level of 0.05.

Organisms constantly interact and transform their environment. As organisms change through time, the ways in which they interact and modify the environment also change (Doebeli and Dieckmann, 2000). Some evolutionary innovations, like photosynthesis, radically transformed the world by increasing oxygen concentration in the atmosphere.

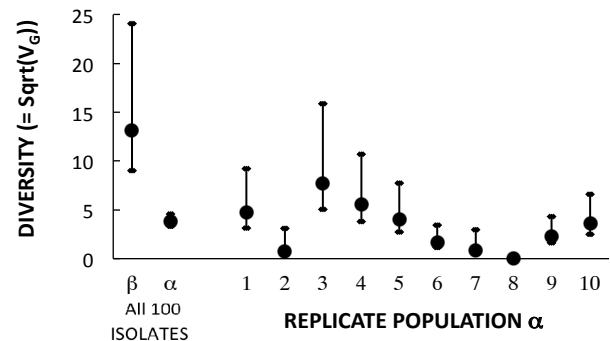


Figure 3. Sources of individual size diversity. Most diversity for individual size arose across replicate populations (β), while within-replicate diversity (α) accounted for less than a third of the explained size variation. Replicate populations differed in the amount of α diversity, with one replicate having no discernable size variation among its ten isolates. Shown are point estimates (square root of the respective genetic variance = $\text{Sqrt}(V_G)$) and 95% CI determined by REML ANOVA.

Here we see that, even within the extremely controlled environment of culture tubes in an incubator, major changes—like the evolution of multicellularity—probably affect environmental conditions. Cluster formation generates spatial structure increasing localized interactions and potentially creates new environmental gradients (Smukalla et al., 2008; Koschwanez et al., 2011). These niche construction dynamics,

as everything else in evolution, are historical processes and the particular history of each of these populations might be also important to account for all the diversity among populations. There was clearly substantial variation in how long multicellularity took to first evolve in the experiment, ranging from one to eight weeks. In addition, other phenotypic changes, like increased cell size, were observed in some populations previous to the appearance of the first multicelled clusters (Ratcliff, et al. 2012;). These observations suggest that chance and environmental changes are playing an important role in this system and together, these factors might account for the high degree of differentiation among populations.

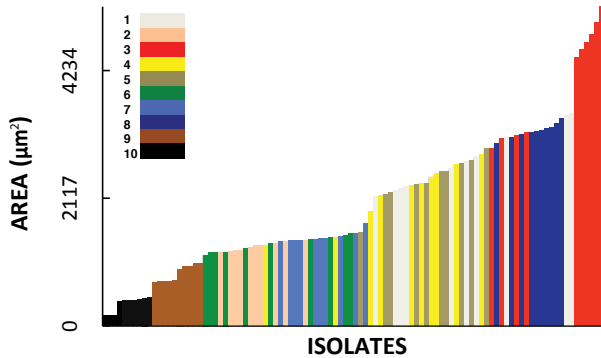


Figure 4. Size distribution of the 100 isolates, ten isolates from each of the ten replicate populations. Isolates are color coded by the source replicate. β diversity (contribution from different replicates) is highest at intermediate phenotypes and declines at phenotypic extremes ($p < 0.0001^*$).

Diversity within each population (Figure 3) is a major contributor of total diversity, albeit smaller than across populations. Phenotypic diversity at extreme sizes is largely localized within a few populations (Figure 4). Moreover, with the exception of population eight, there are at least two different sized genotypes in all populations. Nonetheless, different populations have different degrees of α diversity. Populations one and three have the highest levels of diversity, whereas population two has a low level of isolate differentiation and population eight is monomorphic for cluster size (Figure 3). In this case, in contrast with diversity between populations, different sized snowflake yeast compete for the same resources, making polymorphisms less likely to be explained in terms of pure chance. Furthermore, neutral processes would be unlikely to preserve diversity in an adaptive trait like cluster size in almost all populations.

Thus, there likely are some ecological differences between different genotypes within a population, allowing stable coexistence of these different types. To better understand the causes of within population diversity, we looked closely at one of the populations with higher levels of α diversity (i.e. population one). Within this population there are four distinctive size classes with some overlap among them (Figure 5). Moreover, these differences are heritable (size distribution was maintained after propagation in fresh media each 24 hours for three days without gravitational selection).

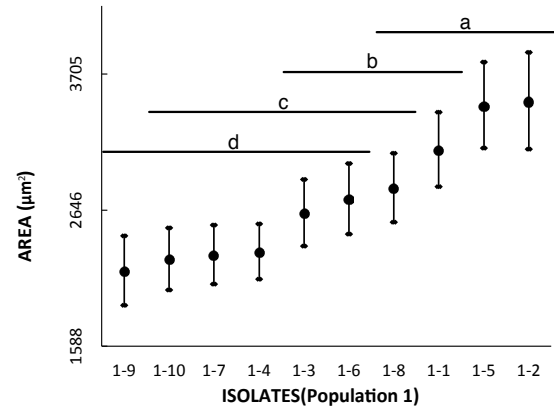


Figure 5. Divergence for individual size readily evolved within replicate populations. Shown are mean sizes and 95% CI for ten genotypes isolated from replicate population one. There are at least four size classes (a through d) based on a Tukey HSD test with a significance level of 0.05

We previously demonstrated (Ratcliff et al., 2012) that there is a trade-off between growth rate and settling and that isolates of different populations fall in different points of that trade-off. Yeast clusters that are fast settlers tend to grow slowly, and *vice-versa*. Thus, we hypothesized that diversity within populations could be explained by a trade-off in cluster size and growth rate, that isolates of the same population would show a negative relation between these two traits. To test this prediction we determined cluster density at time zero, four and eight hours of growth for six isolates of population one (three big and three small).

Source	DF	Sum of Squares	F Ratio	P-value
Early/Late	1	0.0688910	70.2104	<0.0001*
Isolate	5	0.0119729	2.4404	0.0635
Isolate*Early/Late	5	0.0225409	4.5945	0.0044*

Table 1. ANOVA of different effects on growth rate. Differences are significant between early (first four hours) and late growth (four to eight hours of growth) as well as the interaction of time and isolate. Differences between isolates are not significant (see text).

Our results show that, during the first four hours, all the isolates have roughly the same growth rate. Then, after four hours, growth rate decreases in the three biggest isolates. Results of an ANOVA support this conclusion; showing that differences between early (0 to 4 hrs) and late (4 to 8 hrs) growth rates are statistically significant, and more importantly, vary among isolates. Additionally, as it would be expected, differences between isolates are not significant because growth rate is initially very similar. However, later growth rate is different for different isolates and as a result the interaction of terms is significant (Table 1).

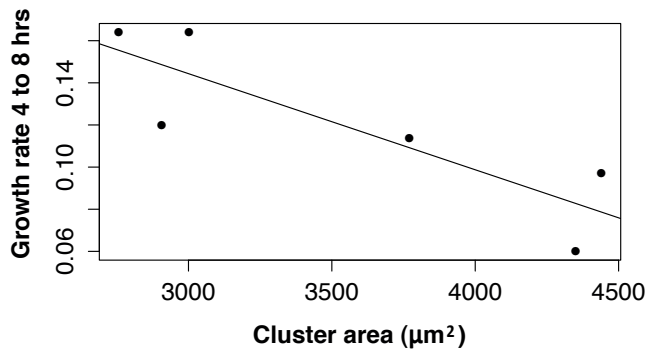


Figure 6. Association of growth rates with individual size. Larger individuals suffer a growth rate disadvantage over the later four hours of growth. There is a -0.85 correlation ($p=0.030$), and a regression would account for 66% of the variation in growth rate among isolates.

Finally, figure 6 shows that within population one there is a trade-off between cluster area and late growth rate (from four to eight hours). Taken together all these observations suggest that there may be within-population ecological differentiation as a result of a trade-off between growth rate and individual size. Some clusters might take advantage of having a faster growth rate whereas others have a bigger size, allowing for a faster settling. This trade-off could, as a result, explain within population diversity, however, more experiments are needed to determine if differentiation along this trade-off allows coexistence.

Conclusion

These results demonstrate that dramatic divergence in individual cluster size readily evolves during the transition to multicellularity. The majority of diversity arises among populations (β), and replicates are readily distinguished by mean individual size. Given the short time scale over which the evolution experiment was carried out (60 serial transfers), it is unclear if the variation for individual size among replicates is likely to persist. Transient diversity can arise via temporal dynamics in the appearance and fixation of different beneficial mutations, prior to convergence to a single adaptive solution. However, multiple lines of evidence suggest that among population diversity is likely to remain. We previously demonstrated a functional trade-off in settling and biomass accumulation (Ratcliff et al., 2012), that persisted over the course of the selection experiment. Here, we demonstrate a similar trade-off of individual size with growth rate, and have observed this variation *within* a single replicate population. While such *within* population variation could potentially be the consequence of simultaneous selective sweeps (clonal interference), the number of genotypes with distinctive phenotypes (four) suggests the evolution of an adaptive radiation (Rainey and Travisano 1998). We suggest that the transition to multicellularity readily promotes the evolution of novelty associated with adaptive radiations.

Acknowledgments. The authors would like to acknowledge Eric Seabloom for statistical advice.

References

- Blount, Z. D., Borland, C. Z., and Lenski, R. E. (2008) Historical contingency and the evolution of a key innovation in an experimental population of *Escherichia coli*. *PNAS* 105(23):7899-7906.
- Bonner, J. T. (1998). The origins of multicellularity. *Integrative Biology*, 1: 27-36.
- Chase, J. M. (2010). Stochastic community assembly causes higher biodiversity in more productive environments. *Science*, 328: 1388-1391.
- Crepet, W. L. and Niklas, K. J. (2009). Darwin's Second "Abominable Mystery": Why are there so many angiosperm species? *Am. J. Botany*, 96(1): 366-381.
- Doebeli, M. and Dieckmann, U. 2000. Evolutionary branching and sympatric speciation caused by different types of ecological interactions. *Am. Nat.* 156 supplement.
- Fraser, C., Alm, E. J., Polz, M. F., Spratt, B. G. and Hanage, W. P. (2009). The bacterial species challenge: Making sense of genetic and ecological diversity. *Science*, 323(5915):741-746.
- Friesen, M. L., Saxer, G., Travisano, M. and M. Doebeli. (2004). Experimental evidence for sympatric ecological diversification due to frequency dependent competition in *Escherichia coli*. *Evolution*, 58(2): 245-260.
- Graham, C. H. and Fine, P. V. A. (2008). Phylogenetic beta diversity: linking ecological and evolutionary processes across space and time. *Ecology Letters* 11(12):1265-1277.
- Genner, M. J. et al. (2004). Beta diversity of rock-restricted cichlid fishes in Lake Malawi: importance of environmental and spatial factors. *Ecography*, 27: 601-610.
- Gould, S. J. (1989). *Wonderful Life: The Burgess Shale and the Nature of History*. Norton, New York, NY.
- Ispolatov, I., Ackermann, M. and Doebeli, M. (2011). Division of labour and the evolution of multicellularity. *Proc. R. Soc. B*, doi:10.1098/rspb.2011.1999
- Kerr, B., Riley, M. A., Feldman, M. W. and Bohannan, B. J. M. (2002). Local dispersal promotes biodiversity in a real-life game of rock-paper-scissors. *Nature*, 418: 171-174.
- Koschwanetz, J. H., Foster, K. R., Murray, A. W. (2011). Sucrose utilization in budding yeast as a model for the origin of undifferentiated multicellularity. *PLoS Biol* 9:e1001122.
- Lenski, R. E., Rose, M. R., Simpson, S. C. and Tadler, S. C. (1991). Long-term experimental evolution in *Escherichia coli* I. Adaptation and divergence during 2,000 generations. *American Naturalist*, 138(6):1315-1341.
- Lenski, R. E. and Travisano, M. (1994). Dynamics of adaptation and diversification: a 10,000-generation experiment with bacterial populations. *PNAS* 91:6808-6814.
- Maynard Smith, J. and Szathmáry, E. (1995). *The Major Transitions in Evolution*. Oxford University Press, Oxford, England.
- Meyer, J. R. et al. (2012). Repeatability and contingency in the evolution of a key innovation in phage lambda. *Science*, 335:428-432.
- Nedelcu, A. M. and Michod, R. E. (2004). Evolvability, modularity, and individuality during the transition to multicellularity in volvocalean green algae. In Schlosser, G. and Wagner, G. P. *Modularity: In Development and Evolution*. University of Chicago Press. Chicago, IL.
- Qian, H., Ricklefs, R.E. and White, P.S. (2005). Beta diversity of angiosperms in temperate floras of eastern Asia and eastern North America. *Ecology Letters*, 8: 15-22.
- Qian, H., Badgley, C. and Fox, D.L. (2009). The latitudinal gradient of beta diversity in relation to climate and topography for mammals in North America. *Global Ecology and Biogeography*, 18:111-122.
- Quian, H. (2009). Beta diversity in relation to dispersal ability for vascular plants in North America. *Global Ecology and Biogeography*, 18:327-332.
- Rainey, P. B. and Travisano, M. (1998). Adaptive radiation in a heterogeneous environment. *Nature* 394:69-72.

- Ratcliff, W. C., Denison, R. F., Borrello, M. and Travisano, M. (2012). Experimental evolution of multicellularity. *PNAS* doi: 10.1073/pnas.1115323109
- Smukalla, S. et al. (2008). FLO1 is a variable green beard gene that drives biofilm-like cooperation in budding yeast. *Cell*, 135:726–737.
- Sterennly, K. (2011). Evolvability reconsidered. In Calcott, B. and Sterennly, K. *The Major Transitions in Evolution Revisited*. Vienna Series in Theoretical Biology, MIT Press, Cambridge, MA.
- Travisano, M., Mongold, J. A., Bennett, A. F. and Lenski, R. E. (1995). Experimental tests of the roles of adaptation, chance and history in evolution. *Science*, 267(5194):87-90.
- Travisano, M. (1996). Long-term experimental evolution in *Escherichia coli*. VI. Environmental constraints on adaptation and divergence. *Genetics* 146:471-479.
- Travisano, M. (2009). Long-term experimental evolution and adaptive radiation. In Garland, T. and Rose, M. R., editors, *Experimental Evolution*, pages 111-134, UC-Press, Berkeley and Los Angeles, California.
- Whittaker, R. H. (1960). Vegetation of the Siskiyou mountains, Oregon and California. *Ecological Monographs*, 30(3):279-338.
- Whittaker, R. J., Willis, K. J. and Field, R. (2001). Scale and species richness: towards a general, hierarchical theory of species diversity. *Journal of Biogeography*, 28: 453-470.

Comparing Distance-Based Phylogenetic Tree Construction Methods Using An Individual-Based Ecosystem Simulation, EcoSim

Ryan Scott¹, Robin Gras¹

¹University of Windsor
scotto@uwindsor.ca

Abstract

Phylogenetic trees are constructed frequently in biological research to provide an understanding of the evolutionary history of the organisms being studied. Often, the actual phylogenetic tree is unknown and the phylogenetic tree constructed is an estimate. There are many methods of phylogenetic tree construction which fall into two main categories: distance-based methods and character-based methods. To test the accuracy of these methods, it is necessary that the system being studied is one for which the actual phylogenetic tree is known. EcoSim is an ecosystem simulation in which predator and prey agents possessing a complex behavioral model can interact, evolve and speciate. In this experiment, we used EcoSim to test the accuracy of the three main distance-based phylogenetic tree construction methods, when constructing a single tree and when performing phylogenetic bootstrapping. Since EcoSim provides data regarding speciation events, we were able to construct the actual phylogenetic trees from this data. We then performed the UPGMA, Neighbor-Joining, and Fitch-Margoliash methods at various time-steps and used symmetric distance as a metric to compare the topologies of the actual and estimated trees. On average, trees contained nearly 30 taxa. We found that the Fitch-Margoliash method with bootstrapping performed slightly better than the other methods, however no method constructed trees in which more than 50% of the partitions were correct.

Keywords: evolution, ecosystem, individual-based model, distance-based, phylogeny, consensus, speciation, phylogenetic bootstrapping.

Introduction

An interesting topic in biology is the construction of phylogenetic trees. Phylogenetic trees are constructed in an attempt to reconstruct the evolutionary past; to develop an understanding of when and which speciation events may have occurred to give rise to the organisms exhibited today. A phylogenetic tree consists of edges, internal nodes, and external nodes (leaves). Leaves represent operational taxonomic units (OTUs) which are the actual species from which data was gathered to construct the tree. The internal nodes are hypothetical taxonomic units (HTUs). They represent the hypothetical last common ancestors to all other species arising from them. The edges often represent the relatedness or genetic distance between two nodes, where a

shorter edge length means species are more closely related. In some trees, edges may be considered an estimation of the time taken between speciation events. In the study of real organisms, constructed phylogenetic trees are often an estimate of the real phylogenetic tree, since the actual phylogenetic tree is usually unknown. Given different data types, there are many different methods that researchers can employ to estimate phylogenetic trees. There are two main groups of phylogenetic tree reconstruction methods: distance-based methods and character-based methods (consisting of subgroups parsimony, compatibility, and maximum likelihood methods) (Felsenstein, 1988).

Distance-based methods could rely on many different types of data to perform analysis including genetic distance from sequences, distances from immunological studies, and Euclidean distance applied in various ways (Wiley and Lieberman, 2011). In terms of distance-based phylogenetic tree construction methods, there are three methods that are more common: Unweighted Pair Group Method with Arithmetic Mean (UPGMA) (Sneath and Sokal, 1973), Neighbor-Joining (NJ) (Saitou and Nei, 1987), and Fitch-Margoliash (Fitch and Margoliash, 1967). Each algorithm has some known properties or cases in which the tree should be very similar to the actual tree. The UPGMA algorithm should produce a correct tree if the distance data is ultrametric, which also means that the evolutionary rates among taxa are constant. This is rarely the case in nature. The Neighbor-Joining algorithm and Fitch-Margoliash method perform well when the distance data is additive. Again, this is usually not the case either. These methods generate a single tree for any given distance matrix. Of the three methods, UPGMA is the most computationally efficient; the algorithm for UPGMA is of complexity $O(n^2)$ (Murtagh, 1984). The Neighbor-Joining algorithm is of complexity $O(n^3)$ (Mailund et al, 2006), and the least efficient of the three, the Fitch-Margoliash method, runs in complexity of $O(n^4)$ (Lespinats et al, 2011). Since distance matrices can be generated from pairwise Euclidean distance data, distance matrices usable in phylogenetic tree construction could be generated using Euclidean distances between points in n -dimensional space. Character-based methods can rely on a variety of phylogenetic characters such as genetic, morphological, behavioral, and molecular attributes to construct phylogenetic trees. Provided that there

is variation among taxa in the attribute and that the attribute is heritable, it could potentially be used as a phylogenetic character (Grandcolas et al, 2001). The characters, if necessary, may be discretized to allow for discrete character states to be generated (Wiley and Lieberman, 2011). The algorithms used to create phylogenetic trees using phylogenetic characters are generally more complex than distance-based methods (Felsenstein, 1988). Generally, these algorithms are based on an optimization criterion such as parsimony, maximum likelihood, or compatibility (Felsenstein, 1988). Character-based methods are quite commonly used in studies of nature, because it is said that data is lost when converting data for use with distance-based methods (Felsenstein, 1988). In this experiment, we focus solely on distance-based methods because we are not dealing with data from a real biological system, we are instead dealing with data that does not contain character-based attributes but instead contains numerical attributes for which it is more appropriate to use distance-based methods. Furthermore, character-based methods tend to be far more computationally complex.

A common practice in phylogenetic tree construction is bootstrapping, in order to test the repeatability of the results (Felsenstein, 1983). Bootstrapping is a resampling method in which the original data is resampled with replacement of characters (Felsenstein, 1983). Bootstrapping allows one to observe in what proportion of trees a particular partition of the tree is represented when data is resampled without removing data. Commonly, a large number (100-1000) of such resamplings are carried out. From these 100-1000 trees generated from bootstrapping, a single tree is generated that contains only the most represented partitions. The generated tree is known as a consensus tree (Felsenstein, 1988). There are several types of consensus tree construction methods, among them the “strict” consensus, “majority rule” consensus, and “majority rule extended” consensus (Felsenstein, 2004). Strict consensus creates a tree consisting only of partitions that were represented in all of the trees (Felsenstein, 2004). Majority rule consensus creates a tree consisting of partitions that occurred more than 50% of the time, but leaves all other partitions unresolved (Felsenstein, 2004). Lastly, majority rule extended creates a tree consisting of partitions that occurred more than 50% of the time, but then it resolves the rest of the tree by using the most represented partitions (Felsenstein, 2004). It is possible to calculate distances between trees, though there are many methods of doing so which are not verified in terms of accuracy. Furthermore, of those that have been verified, many are situational. The “symmetric distance” is a metric useful for determining distances between trees pertaining to topology, without considering branch lengths (Felsenstein, 2004). It is also a quite simple algorithm. If given two trees, you can simply count the number of partitions which do not exist in the other tree. This metric is useful because there is a maximum distance between two trees. Between two trees containing n taxa, the maximum distance is $2n-6$. Therefore, these symmetric distance values are subject to normalization by dividing all values by $2n-6$. Tree with a normalized symmetric distance of 1 are trees that share no

partitions, and trees with a normalized symmetric distance of 0 are identical.

Researchers regularly attempt to create new methods or improve old ones, but little is known about what factors may determine which method is the best. In order to determine which factors favor which method, a study using a simulation would be intriguing, because a large amount of data could be generated very quickly, and the actual phylogenetic trees would be known. Thus, comparisons between the actual trees and the estimated trees could be made. The purpose of our experiment is to determine the accuracy of various distance based tree construction methods with and without bootstrapping. As we are most interested in tree topology and would like the ability to normalize tree distance values to allow comparison of results between different generations, symmetric distance is our distance of choice. This experiment requires a system from which a large amount of meaningful data can be efficiently acquired, and most importantly, for which the actual phylogenetic tree is known. Further, the conclusion of an experiment conducted by Hang et al (Hang et al, 2007) and Hagstrom et al (Hagstrom, et al, 2004) is that computer simulations often underestimate the accuracy of phylogenetic methods due to the non-existence of natural selection. Therefore, a system in which natural selection exists would be most valuable. For this experiment, our system of choice is EcoSim because like Avida, it exhibits natural selection, efficiently produces meaningful data, and tracks phylogenetic records.

The Ecosystem Simulation, EcoSim

EcoSim is an individual-based predator-prey ecosystem simulation in which agents can evolve (Gras et al, 2009). The agents have a behavior model which allows the evolutionary process to modify the behaviors of the predators and prey. Furthermore, there is a speciation mechanism which allows researchers to study global patterns as well as species-specific patterns. To our knowledge, EcoSim is the only simulation in which agent behaviors affect evolution and speciation. In EcoSim, an individual's genomic data codes for its behavioral model and is represented by a fuzzy cognitive map (FCM) (Kosko, 1986). The FCM contains sensory concepts such as foodClose or predatorClose, internal states such as fear or hunger, and motor concepts such as escape or reproduce. The FCM is represented as a 390-element array consisting of positive or negative floating-point values which represent the extent to which one concept influences another. For example, it would be expected that the sensory concept predatorClose would positively affect the internal concept fear, which would then positively affect the escape motor concept. Likewise, sensing that a predator is close should negatively affect hunger, which should result in a prey agent choosing not to eat when a predator is too close. Of course, these relationships among concepts evolve over time, sometimes giving a new meaning to a concept. This representation of the FCM allows for reasonable computational complexity while still allowing for a complex system with meaningful genomic information. Furthermore, the FCM is heritable, meaning that a new agent

is given an FCM which is a combination of that of its parents with possible mutations. The FCM is largely responsible for the evolution, speciation, and behavior model which makes EcoSim so unique. EcoSim subscribes to the “genotypic cluster” definition of a species, which states that “species are clusters of genotypes circumscribed by gaps in the range of possible multilocus genotypes between them” (Mallet, 1995). What this means, in EcoSim, is that if the difference between FCMs of the two most dissimilar conspecific individuals is greater than a set threshold, the species will then split and the new species will be reproductively isolated from the parent species (Aspinall and Gras, 2010). Each species of EcoSim is assigned a species ID, which is simply a count of how many species have existed in that run (starting at species 1). Thus, species 1 is the common ancestor of all other species in a run. All trees produced in this experiment (both actual and estimates) refer to species by their species ID. Since EcoSim has the capacity to allow speciation events to occur, it is possible to track speciation events throughout a run of the simulation and construct the actual phylogenetic tree. This is important because it offers us the opportunity to perform various tree reconstruction methods and compare the results with the actual tree, which is generally not possible with real data from biological systems. Since EcoSim uses an array of 390 floating-point values to represent an agent's genome, we can obtain the average FCM of any species at any time step in any particular simulation run. From this data, we are able to construct a pairwise distance matrix of all species alive any particular time step. Thus, we are able to perform and test distance-based phylogenetic tree construction methods on data generated by EcoSim. There have been several other studies conducted using EcoSim. EcoSim has been shown to have realistic species abundance patterns (Devaurs et al, 2010) and chaotic behavior with multi-fractal properties which has been observed in biological systems (Golestani and Gras, 2010). Another study observed disease diffusion patterns and disease control regimes in EcoSim (Farahani et al).

Data Preparation and Phylogenetic Methods

Five EcoSim runs of lengths 5658, 7098, 10000, 15500, and 19500 generations were carried out. The lengths of these runs are arbitrary and do not affect the results. These runs exhibited various run-specific characteristics. Respectively, the aforementioned EcoSim runs had an average global population of about 288740, 216320, 163675, 128530, and 149177 agents, and an average species count of 28.4, 16.3, 36.2, 30, and 29.4 species over the generations which we analyzed. Their average normalized symmetric distances (considering all phylogenetic construction methods) were 0.46, 0.48, 0.66, 0.59, and 0.54, respectively. The species population sizes ranged from 1 to 73242 over all of the runs. On average, there were 29.52 taxa per generation, ranging from 7 taxa to 47 taxa. Thus, the largest distance matrix from which a tree was constructed was 47x47. In this case, to calculate a single tree using the UPGMA or Neighbor-Joining method required less than one second, whereas when using the Fitch-Margoliash method it required nearly ten seconds.

Even if the system has to handle hundreds of thousands of “intelligent” agents simultaneously, the overall complexity of the algorithm is linear and therefore it allows us to compute a very high number of time steps giving us the possibility to observe evolutionary phenomena. For reference, a run of 25000 generations of EcoSim takes approximately 40 days, but this depends on the number of predator and prey individuals produced.

A program was created to automatically generate phylogenetic trees in NEWICK format (Felsenstein, 2004) by extracting data regarding species splitting events from the simulation. The branch lengths of the trees generated by this program were exactly the number of generations passing between speciation events. Another program was implemented to edit the full phylogenetic trees, removing all species that did not exist at a given generation. The purpose of this was to generate actual trees that were comparable with results from the distance-based tree construction methods. Another program was then created to extract species-specific average FCMs at a given generation, and with that data construct distance matrices. This program used pairwise Euclidean distance between average FCMs to generate distance matrices. When analyzing biological systems, one would first have to convert the data (genetic or molecular sequences, enzyme binding data, or immunological data for example) into distance matrices. In the case of molecular or genetic sequences, one would first have to align the sequences and then calculate the genetic or molecular distance between them. Once this is completed, the distance-based phylogenetic tree construction methods can be applied.

Once these distance matrices were generated, the program “Neighbor” of PHYLIP (the PHYLogeny Inference Package) (Felsenstein, 1989) was used to perform Neighbor-Joining and UPGMA methods on the distance matrices. To perform the Fitch-Margoliash method, “Fitch” of PHYLIP was used. For a run of 10000 generations (for which 19000 trees are generated when performing phylogenetic bootstrapping), to compute all of the bootstrap Neighbor-Joining and UPGMA trees it only took about two hours, whereas to compute the bootstrap Fitch-Margoliash trees it took roughly ten hours. The trees generated from these algorithms were compared with the actual trees using symmetric distance. This was done using “TreeDist” of PHYLIP. In order to perform bootstrap analysis, another program was created to resample the FCM and generate distance matrices from these resampled FCMs. This was performed by choosing a replacement probability and then possibly replacing an FCM element with another for all species before calculating distances between species. The assigned replacement probability was 0.5, and 1000 such resamplings were performed. Then, “Consense” of PHYLIP was used to perform majority rule extended consensus. Majority rule extended was used as the consensus method because it generates fully resolved binary trees to allow for comparison with the actual phylogenetic trees. The consensus trees were then compared with the actual trees (again, using “TreeDist”). Tree construction (both the actual trees and distance-based estimates), consensus, and comparisons were

performed every 500 generations until the end of an EcoSim run, starting from a point in the run at which there were enough species in existence for it to be reasonable to test. This resulted in 100 analyzed time-steps, with 3006 tree estimates constructed per time-step.

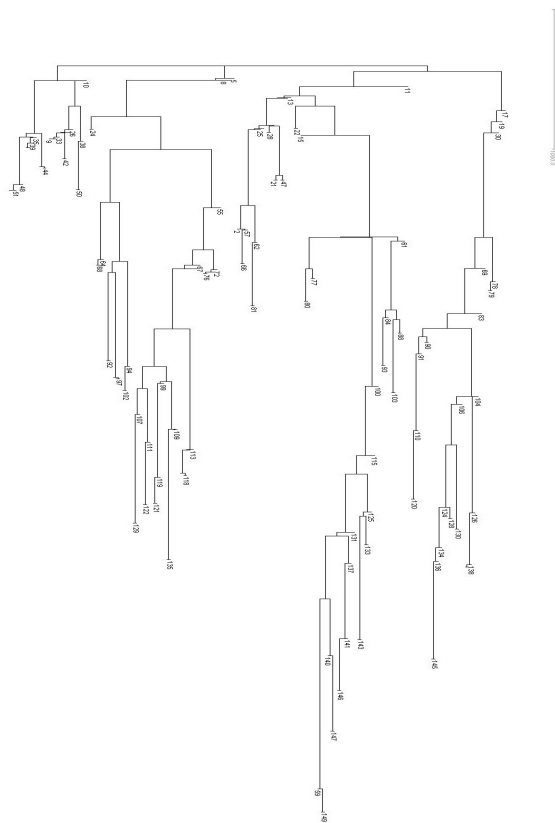


Figure 1: The actual phylogenetic tree for EcoSim run #2 of 5658 generations. Over the 5658 generations, 149 taxa were generated. The leaves of the tree represent the species indicated at the time of the last splitting event in which they were involved. The internal nodes of the tree are the species with the lowest species ID in the partition to the right of that node (since species are given an ID in the order in which they are generated), and represent that particular species at the time of that splitting event.

Results

Actual phylogenetic trees consisting of all species in a run were constructed for all five EcoSim runs, an example of which is shown in Figure 1. Edited trees, consisting of only species existing in a particular generation, were also created. Neighbor-Joining, UPGMA, and Fitch-Margoliash methods were used, and consensus trees using these methods were generated as well. Examples of each tree are shown in Figure 2. The UPGMA method is the only method of the three which creates a binary rooted tree when not performing consensus analysis. While not performing consensus analysis, Neighbor-

Joining and Fitch-Margoliash methods create unrooted trees. All consensus trees are binary rooted trees. The trees produced by performing consensus analysis have branch lengths which are meaningless in terms of evolutionary distance between species. The branches of the consensus trees are actually the bootstrap value; this is number of trees in which the partition to the right of that branch was represented out of the 1000 resamplings performed. Thus, the longer the branch, the more represented that partition was. The tree distance metric we used, as previously mentioned, only deals with topology, so the branch lengths (in terms of comparison) are not necessarily important.

Symmetric distances between the edited actual trees and the estimated trees were calculated (Table 1). Ranked from most effective to least effective, the phylogenetic tree construction methods are as follows: 1) Fitch-Margoliash Consensus, 2) Fitch-Margoliash, 3) Neighbor-Joining Consensus, 4) UPGMA Consensus, 5) UPGMA, and 6) Neighbor-Joining. Note that although it was the most accurate, the Fitch-Margoliash method only classified, on average, 46% of the partitions.

Method	Avg. SD	SD Std. Dev.	Avg. Norm. SD	Norm. SD Std. Dev.
F-M (C)	28.98	11.41	0.54	0.15
F-M	29.24	11.5	0.55	0.15
N-J (C)	29.57	11.52	0.55	0.15
UPGMA (C)	29.86	12.43	0.56	0.16
UPGMA	31.23	13.02	0.59	0.18
N-J	32.44	11.92	0.6	0.14

Table 1: The average and standard deviation of the symmetric distance (SD) and the normalized symmetric distance of all five EcoSim runs. The Fitch-Margoliash method generated the most accurate trees, with an average of 54% of partitions incorrectly reconstructed. The least accurate was the Neighbor-Joining method, with an average of 60% of partitions incorrectly reconstructed. The UPGMA method produced the most varying results, and the Neighbor-Joining method was the most consistent.

Conclusions

In our experiments based on data generated by our evolving ecosystem simulation, none of the distance-based methods performed well. None of the methods, on average, estimated over 50% of the partitions of the trees correctly. Though it is possible that these methods are just not as accurate as previously perceived, there could be several reasons why they performed poorly. It is possible that there are factors (such as mutation rates, small population sizes for some species, rate of evolution, probability of back-mutation, or large number of species) that make it difficult for distance-based phylogenetic tree construction methods to properly recreate the trees. It is also possible that Euclidean distance (employed in this manner) is just a poor metric for use with distance-based phylogenetic tree construction methods. Another possibility is that the distance matrices produced were not additive (and

thus not ultrametric either), but this is often the case in nature as well (Felsenstein, 2004). Lastly, rather than using the entire FCM, it may be better to choose specific FCM values to create phylogenies from, despite research in phylogenomics that

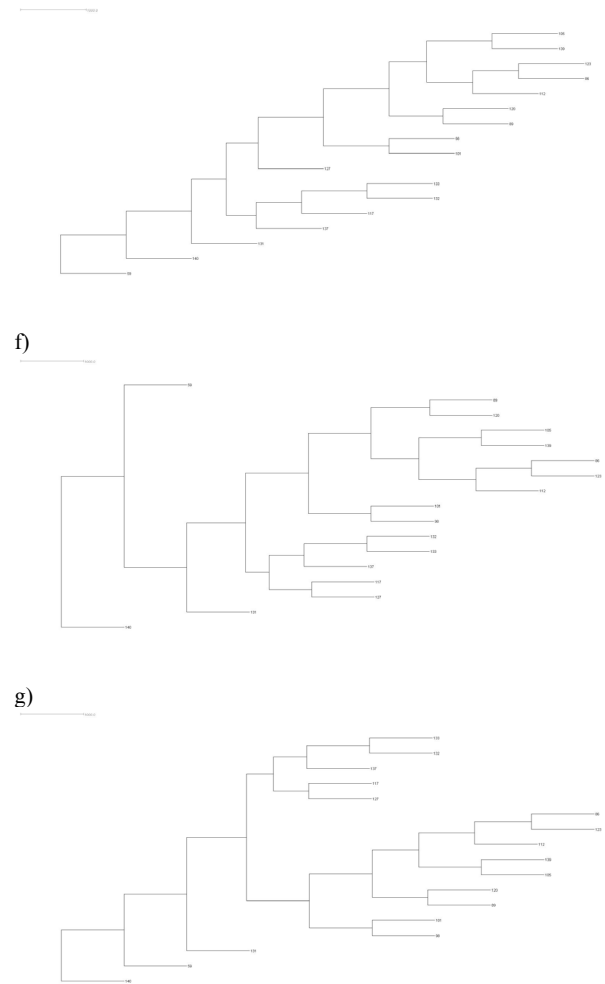
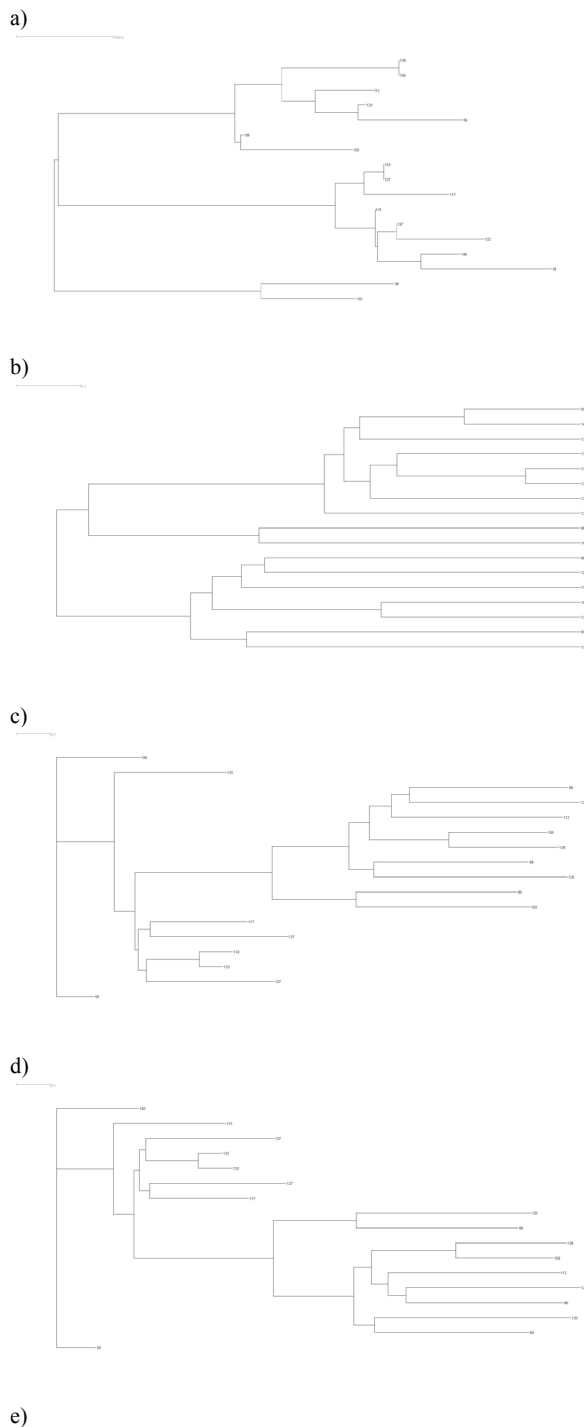


Figure 2: The actual (a) and estimated (b-g) trees for EcoSim run #2, generation #4158. Consensus trees (UPGMA (e), Neighbor-Joining (f), Fitch-Margoliash (g)) and UPGMA (b) trees are all binary rooted trees, while Neighbor-Joining (c) and Fitch-Margoliash (d) trees are unrooted. This example shows the similarities and differences between relatively small (17 taxa) trees generated by the various methods.

suggests using entire genomes (rather than a small number of genes) increases the phylogenetic signal-to-noise ratio (Phylippe et al, 2005; Snel et al, 2005). This is because our FCM may actually be noisy in terms of the phylogenetic data it generates, so determining and focusing on values with high phylogenetic signal-to-noise ratio may increase the accuracy. The Fitch-Margoliash method with consensus analysis performed slightly better than the other methods. It was expected that in all cases, performing phylogenetic bootstrapping and building consensus trees increased the accuracy of the methods.

Our results contrast from those of Hagstrom et al (Hagstrom et al, 2004), as in their experiments they have found that these methods are quite accurate (in many cases,

reproducing the exact phylogenetic tree) provided that there is an element of natural selection in the employed system. EcoSim is such a system, yet our results are quite different. One important difference between these experiments is that in our experiment, we attempted recreating phylogenies consisting of many (on average 29.52) taxa whereas in that of Hagstrom et al, phylogenies of only four taxa were reconstructed.

A study by Leitner et al (Leitner et al, 1996), in which researchers performed various phylogenetic tree construction methods on HIV-1 molecular data, also found that the Fitch-Margoliash method was most accurate, and it also found that Neighbor-Joining consensus was more accurate than UPGMA (though they considered branch lengths in their tree comparison, which may have increased the inaccuracy of UPGMA). They found that in some cases the true phylogeny was successfully reconstructed, whereas in all of our cases this did not occur. It is interesting to note, however, that they only had 9 taxa to analyze. Our best scenario was one in which we had only 7 taxa to analyze, which gave us 25% dissimilarity using Fitch-Margoliash and Neighbor-Joining, and 75% dissimilarity using UPGMA. On average, 29.52 taxa per generation were analyzed in our experiment. It is also interesting to note that choice of gene, in the case of HIV-1, accounted for an average symmetric distance difference of about 25%. This also leads us to believe that perhaps we should focus on specific FCM values (such as those that rapidly evolve or those that are most selected upon) rather than on the entire FCM. When considering the efficiency of the algorithms, the UPGMA and Neighbor-Joining methods are much more efficient than the Fitch-Margoliash method, so it may still be more appropriate to use Neighbor-Joining or UPGMA instead of Fitch-Margoliash in some cases (for example, those that require the computation of many trees).

In the future, we will attempt to determine which characteristics (for example relatedness of different species, speciation threshold, or rates of evolution) may allow each method to produce the most accurate tree. Furthermore, it would be intriguing to determine if these factors lead to better trees overall. We would also like to discover if selecting only certain FCM values produces better trees. It also may be interesting to discretize the FCM values and perform a similar analysis of the more popular character-based methods.

Acknowledgements

This work is supported by the NSERC grant ORGPIN 341854, the CRC grant 950-2-3617 and the CFI grant 203617 and is made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET, www.sharcnet.ca).

References

Aspinall, A., and Gras, R. (2010). K-means clustering as a speciation mechanism within an individual-based evolving predator-prey

- ecosystem simulation. *Active Media Technology*, pages 318–329, Toronto, Canada.
- Devaurs, D., and Gras, R. (2010). Species abundance patterns in an ecosystem simulation studied through Fisher's logseries. *Simulation Modelling Practice and Theory*, 18(1), 100-123.
- Farahani, Y. M., Khater, M., and Gras, R. (in press). Modeling Epidemic Spread in a Predator-Prey Evolutionary Ecosystem Simulation. To appear in the *Journal of Artificial Life*.
- Felsenstein, J. (1985). Confidence limits on phylogenies: An approach using the bootstrap. *Evolution*, 39:783–791.
- Felsenstein, J. (1988). Phylogenies From Molecular Sequences: Inference and Reliability. *Annual Review of Genetics*, 22:521-565.
- Felsenstein, J. (1989). PHYLIP - Phylogeny Inference Package (Version 3.2). *Cladistics*, 5: 164-166.
- Felsenstein, J. (2004). *Inferring Phylogenies*. Sinauer Associates, Inc., Sunderland, MA.
- Fitch, W. M. and Margoliash, E. (1967). Construction of phylogenetic trees. *Science*, 155(3760):279-284.
- Golestani, A., and Gras, R. (2010). Regularity analysis of an individual-based ecosystem simulation. *Chaos (Woodbury, N.Y.)*, 20(4), 043120.
- Grandcolas, P., Deleporte, P., Desutter-Grandcolas, L., and Dageron, C. (2001). Phylogenetics and Ecology: As Many Characters as Possible Should Be Included in the Cladistic Analysis. *Cladistics*, 17:104-110.
- Gras, R., Devaurs, D., Wozniak, A., and Aspinall, A. (2009). An individual-based evolving predator-prey ecosystem simulation using a fuzzy cognitive map as the behavior model. *Artificial Life*, 15(4), 423-63.
- Hagstrom, G. I., Hang, D. H., Ofria, C., and Torng, E. (2004). Using Avida to Test the Effects of Natural Selection on Phylogenetic Reconstruction Methods. *Artificial Life 10*, pages 157-166. MIT Press, Cambridge, MA.
- Hang, D., Torng, E., Ofria, C., and Schmidt, T. M. (2007). The effect of natural selection on the performance of maximum parsimony. *BMC Evolutionary Biology*, 7:94.
- Kosko, B. (1986). Fuzzy cognitive maps. *International journal of man-machine studies*, 24(1), 65-75.
- Leitner, T., Escanilla, D., Franzen, C., Uhlen, M., and Albert, J. (1996). *Proceedings of the National Academy of Sciences of the United States of America*, 93:10864-10869.
- Lespinais, S., Grando, D., Marechal, E., Hakimi, M., Tenaillon, O., and Bastien, O. (2011). How Fitch-Margoliash Algorithm can Benefit from Multi Dimensional Scaling. *Evolutionary Bioinformatics*, 7:61-85.
- Mailund, T., Brodal, G. S., Fagerberg, R., Pedersen, C. N. S., and Phillips, D. (2006). Recrafting the neighbor-joining method. *BMC Bioinformatics*, 7:29.
- Mallet, J. (1995). A species definition for the modern synthesis. *Trends in Ecology & Evolution*, 10:294–299.
- Murtagh, F. (1984). Complexities of hierarchic clustering algorithms: state of the art. *Computational Statistic Quarterly*, 1(2):101-113.
- Phylippe, H., Delsuc, F., Brinkmann, H., and Lartillot, N. (2005). Phylogenomics. *Annual Review of Ecology, Evolution, and Systematics*, 36:541-562.
- Saitou, N. and Nei, M. (1987). The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4:406-425.
- Sneath, P. H. A. and Sokal, R. R. (1973). *Numerical Taxonomy*. Freeman, San Francisco, CA.
- Snel, B., Huynen, M., Dutilh, B. E. (2005). Genome Trees and the Nature of Genome Evolution. *Annual Review of Microbiology*, 59:191-209.
- Wiley, E. O. and Lieberman, B. S. (2011). *Phylogenetics: Theory and Practice of Phylogenetic Systematics*. John Wiley & Sons, Hoboken, NJ.

Evolved Modular Epistasis in Artificial Organisms

Sergi Valverde^{1,2}, Ricard V. Solé^{1,2,3} and Santiago F. Elena^{3,4}

¹ICREA-Complex Systems Lab, Universitat Pompeu Fabra, Dr Aiguader 88, 08003 Barcelona, Spain

²Institut de Biologia Evolutiva (CSIC-UPF), Passeig Marítim de la Barceloneta, 37-49, 08003 Barcelona, Spain

³Santa Fe Institute, 1399 Hyde Park Road, Santa Fe NM 87501

⁴Instituto de Biología Molecular y Celular de Plantas, Consejo Superior de Investigaciones Científicas-UPV, 46022 València, Spain

E-mail: sergi.valverde@upf.edu

Abstract

How does complexity evolve in artificial and natural systems? A central concept within genetic systems is epistasis, namely the modulation of the effects of a given gene by one or several other genes. Epistasis is known to have an impact on many features of organisms, from recombination and sex to the ruggedness of the underlying fitness landscapes. However, the multi-scale nature of evolution and organisms makes often difficult to properly characterize epistatic interactions. Here we study the hierarchical organization of epistatic interactions between machine instructions in evolved digital organisms. We present a new quantitative approach to discover epistatic interactions that is able to capture the presence and role of groups of epistatic modules. Therefore, it thus takes into account the intrinsic nested nature of individual complexity. We found evidences of modular epistasis in avidians, with some modules having a tendency toward antagonistic epistasis while others show the opposite epistatic sign. We also found that this modular organization was correlated to organismal robustness.

Introduction

Genetic interactions and their impact on phenotypic traits are known to define a nonlinear mapping, which strongly affect evolutionary trajectories (Kauffman, 1993). Such nonlinear character of gene interactions is often named as *epistasis* (Van Driessche et al, 2005; Sanjuán and Elena, 2006; Collins et al., 2007, Zheng et al., 2010, Elena et al., 2010). We can understand the functional role of any component by looking at the consequences of perturbing it. Unfortunately, the above approach is limited and cannot reconstruct the functional organization of systems with ambiguous phenotype-genotype mappings. For example, we will not observe any phenotypic change if we perturb one out of two redundant components. In this context, we can extend single-perturbation experiments to double-perturbation experiments that discard redundancies explicitly.

Epistatic interactions have been used to detect functional associations between pairs of genes. Non-scaled epistasis among a pair of mutations i and j is defined as

$$\epsilon_{ij} = W_{ij} - W_i W_j \quad (1)$$

where W_i and W_j represent the fitness values of single mutants and each entry of the matrix W_{ij} indicates the fitness value of the corresponding double mutant. Depending on the value of the above we have three different types of interactions: (1) no epistasis when $\epsilon_{i,j} = 0$, (2) synergistic epistasis when $\epsilon_{i,j} < 0$ and (3) antagonistic epistasis when $\epsilon_{i,j} > 0$.

The analysis of complex biological systems suggests that interactions between components take place between multiple scales and in the presence of feedback loops. This makes functional reconstruction a challenging and time-consuming task. In principle, the previous definition can be naturally extended to consider multiple associations, *i.e.*

$$\epsilon_{i_1, i_2, \dots, i_n} = W_{i_1, i_2, \dots, i_n} - \prod_{\mu=1}^n W_{i_\mu} \quad (2)$$

but the required testing, involving multiple knockouts is, however, very costly and becomes rapidly intractable. Within the context of regulatory gene networks, it has been shown the presence of complex interactions between epistasis, network redundancy and degeneracy (Macía et al., 2012). Similarly, Lenski et al. (1999) found that epistasis was predominantly synergistic for complex digital organisms but switched to mostly antagonistic for simpler organisms.

Interestingly, the more complex organisms were also more robust against the effect of mutations than the simpler ones. The difficulties for reaching a proper understanding of the role played by epistatic interactions within complex networks calls for novel approximations. Here, we propose a new, efficient, multi-scale analysis of epistatic interactions to uncover the so-called "epistatic modules", that is, groups of related instructions and functions with similar epistatic interactions. Such approach can be useful to better understand the emergence and organization of epistasis interactions between different subcomponents of evolved organisms.

Methods

Our model organisms are digital creatures evolved within the Avida system (Ofria and Wilke, 2004). This has several

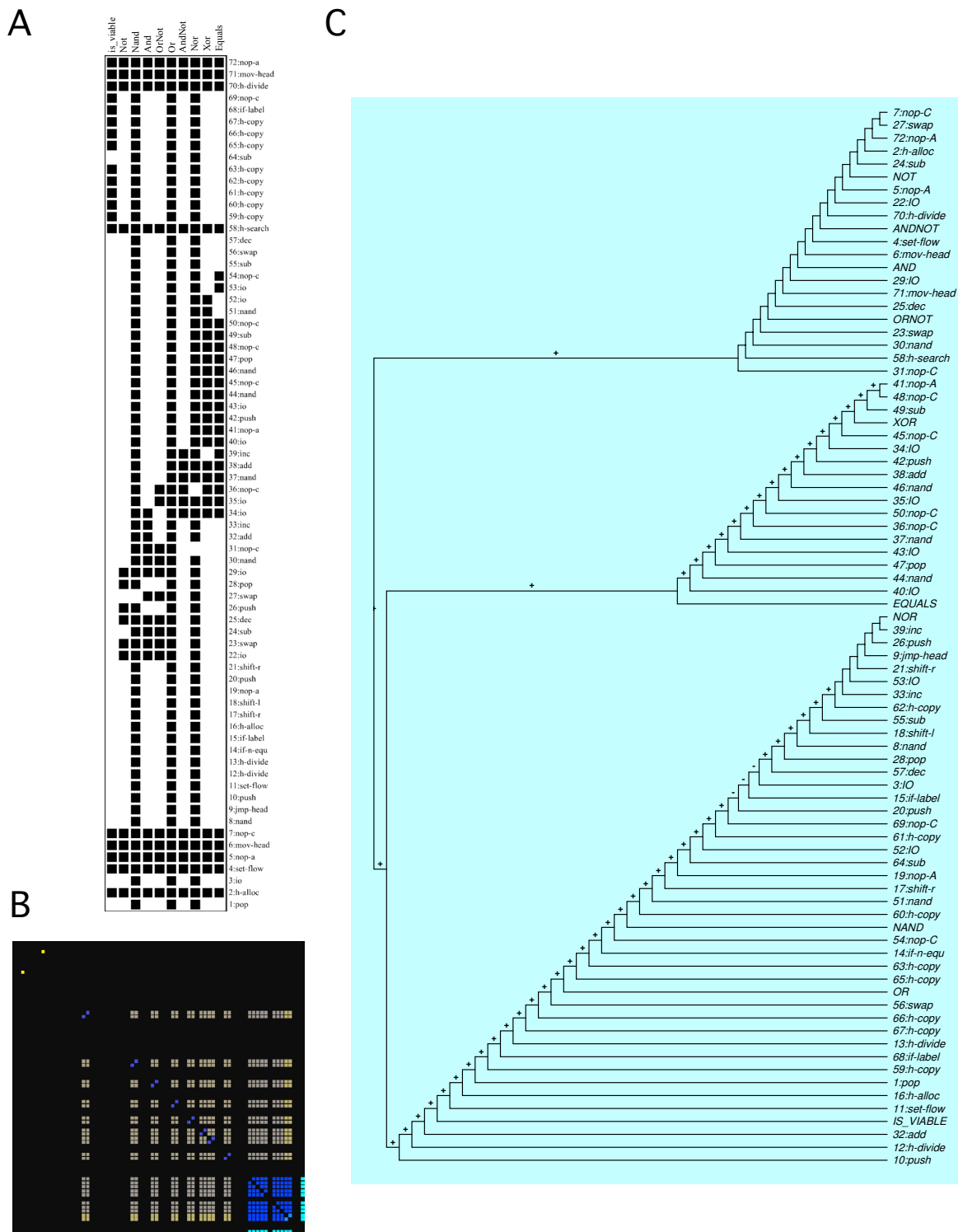


Figure 1: Analysis of functional modularity for a brittle avidian ($\langle \epsilon_{\omega} \rangle > 0$) with high modularity ($Q = 0.21$). (A) Task map showing the implication of each genomic instruction on the nine different tasks. (B) Heat-map illustrating the intensity of epistatic interactions between pairs of instructions in the genome. The stronger the blue, the more antagonistic (positive) epistasis; the stronger the yellow, the more synergistic (negative) epistasis. (C) Cladogram constructed from the epistasis matrix. Branches have been decorated with the average epistasis of the corresponding subtree (see text).

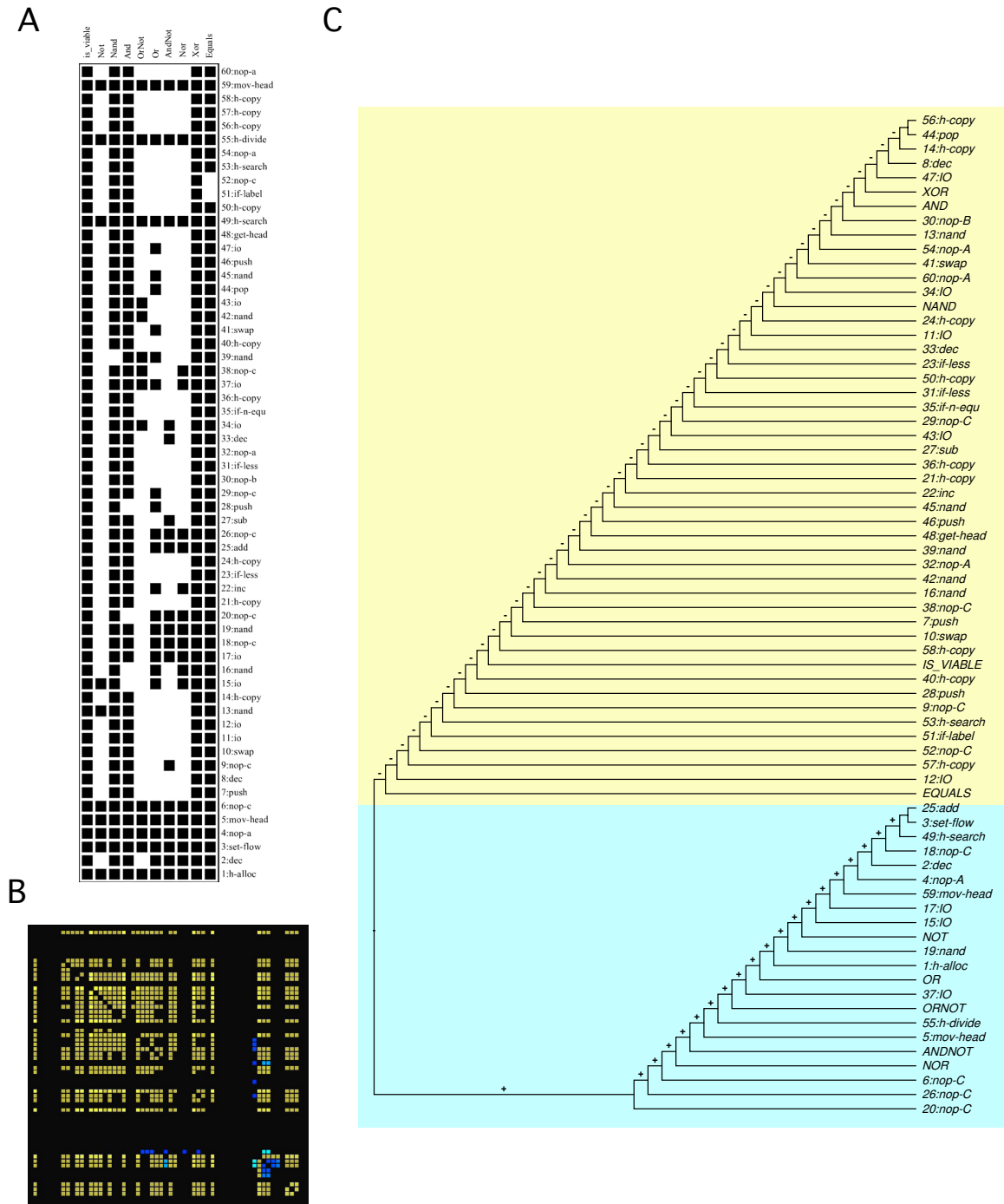


Figure 2: Analysis of functional modularity for a robust avidian ($\langle \epsilon_{\omega} \rangle < 0$) with low modularity ($Q = 0.15$). (A) Task map showing the implication of each genomic instruction on the nine different tasks. (B) Heat-map illustrating the intensity of epistatic interactions between pairs of instructions in the genome. The stronger the blue, the more antagonistic (positive) epistasis; the stronger the yellow, the more synergistic (negative) epistasis. (C) Cladogram constructed from the epistasis matrix. Branches have been decorated with the average epistasis of the corresponding subtree (see text).

advantages: (1) we can readily simulate many organisms in very different conditions and (2) for each artificial organism (or 'avidian') we have a clear correspondence between its genome (instructions) and the different logic tasks solved by the organism (its phenotype). An avidian consists of a CPU, a memory that stores the 'genome', registers and input/output buffers. The genome is described with a program consisting of different instructions to be interpreted by the CPU to perform different actions. The Avida system rewards any digital organism that computes a pre-defined repertoire of nine target high-level tasks. For each avidian, we also obtain a representation of the phenotype-genotype (or task) map $A_{ij} = 1$ if j -th task depends on the i -th instruction to be completed and $A_{ij} = 0$ if they are independent.

As it is illustrated in Fig. 1A and Fig. 2A, the same instruction can be involved in the implementation of more than one task. For example, visual inspection of the task-map in Fig. 2A indicates there are clusters of instructions with similar behavior, i.e., mutations in these instructions tend to affect the same subset of functions (e.g., instructions 22-29). Many natural and artificial networks display modular organization, that is, there are subgroups of nodes (also called modules or communities) significantly more connected between them than with the rest of nodes. Also, we can look at the pattern of connections exchanged between these modules or at the internal structure of modules (e.g., modules-within-modules).

Intuitively, we can see the modular organization as a partition of the network in distinct subparts (see below). Module detection is not an easy task because the genotype-phenotype mapping is not typically a one-to-one relationship. Here, we use the following mathematical approach to systematic module detection, in other words, disentangling the phenotype-genotype mapping. The task map A_{ij} is formally a bipartite network having two types of nodes, e.g., instructions (genotype) and functions (phenotype). This is a particular class of networks satisfying the property that there are no links between nodes of the same type, that is, interactions between functions are indirect and always mediated through, at least, one instruction. Module detection in bipartite networks is equivalent to the maximization of the so-called *modularity* (Newman and Girvan, 2004), which is an heuristic measure of the quality of any modular partition of the network:

$$Q = \frac{1}{m} \sum_{i,j} [A_{i,j} - P_{i,j}] \delta(g_i, g_j) \quad (3)$$

where $m = \sum A_{i,j}$ is the total number of links, $P_{i,j} = k_i k_j / m$ is the probability that instruction i and function j are related (this takes into account the density of the task map), node i has been assigned to module g_i and $\delta(x, y) = 1$ if $x = y$ or $\delta(x, y) = 0$, otherwise. Notice that this definition of modularity is different from those previously proposed by Misevic et al. (2006) to analyze the evolution of

physical and functional modularity in avidians as a result of sexual reproduction. Here, high values of Q correspond to highly modular partitions of the task map. In this case, instruction i and function j are classified in the same module so $g_i = g_j$ (and thus $\delta(g_i, g_j) = 1$) because the difference $A_{i,j} - P_{i,j} > 0$ is a large value.

The bipartite modularity algorithm finds the partition (i.e., the g_i mapping) that maximizes the Q value (Barber, 2007). From the computational point of view, the bipartite modularity algorithm is roughly equivalent to finding a hierarchical decomposition (a cladogram) of the network, that is, modular structure corresponds to a natural hierarchy of groups (and sub-groups) of instructions and functions (see Fig. 1C and Fig. 2C). We have found that the modules obtained with our algorithm have a functional meaning. Complex organisms might display a hierarchical organization of epistatic modules, where complex functions depend on simpler functions implemented at lower levels.

To better understand the relationship between modular organization and epistasis, we evaluated the sign of average epistasis for each module (and sub-module) defined in the cladogram. To do so, we compute the average epistasis for each node in the cladogram as:

$$\langle \epsilon_\omega \rangle = \frac{1}{S_\omega} \sum_{i,j \in \omega} \epsilon_{ij} \quad (4)$$

where ω is the subset of all the instructions (e.g., the tips) in the node subtree, and ϵ_{ij} is the pairwise epistasis between tips at the lowest level of the cladogram (Eq. 1). We have implemented a new analytical tool in Avida to generate the epistasis matrices shown in Fig. 1B and Fig. 2B.

Results and Discussion

To illustrate how the algorithm works, we show the results of applying it to two avidians that were evolved to differ in their robustness against mutational perturbations (Elena and Sanjuán, 2008). In this example, the impact of changes depends on the complexity of the function associated to the mutated instructions. In agreement with previous results (e.g. Lenski et al., 1999; Edlund Adami, 2004; Elena et al., 2007; Elena and Sanjuán, 2008), the more robust avidian was built in such a way that average epistasis is synergistic, whereas the brittle one shows a predominance of positive epistasis. However, our algorithm shows that the situation is not as simple as the average values may suggest, since both types of organisms, robust and brittle, are build up with modules of varying epistatic signs. Indeed, brittle organisms are typically more modular although modules show epistasis of both signs (Fig. 1C). By contrast, more robust organisms are typically less modular with an abundance of antagonistic interactions, yet containing modules dominated by synergistic epistasis (Fig. 2C).

Epistasis plays a crucial role in defining and modeling evolutionary dynamics of gene interactions and genomes. It

provides a well-defined, quantitative framework to analyze the nature and complexity of genotype-phenotype maps. Given the difficulties associated with its standard definition (Eq. 2) and under the assumption that evolved organisms involve multiple levels of nested complexity, we have proposed a network-based method to study the relationship between epistasis and modularity in artificial organisms. Such measure captures the modular nature of epistatic interactions and thus properly characterizes the internal structure of digital organisms and how they evolve and more complex, robust architectures.

Modular epistasis, that is, the situation when functional modules are constituted by genes involved epistatic interactions of a given sign, seems to be a pervasive property of biological systems (*e.g.*, Segrè et al., 2005; Costanzo et al., 2010; He et al., 2010; Xu et al. 2011). Our results suggest that selection for robustness may favor asexuals which have more modules, with variance among modules in the type of epistasis they have, although showing an overall synergistic epistasis. Relaxation of the selection for robustness favors more modular organisms with an overall antagonistic epistasis, although the existing modules still may vary on the sign of epistasis. In ongoing work, we are generating extensive data resulting from the application our novel methodology to populations of asexuals evolved under different genetic (robust/brittle, sexual/asexual) and environmental conditions (constant/varying environments) and will infer some generalities about the origin of genomic architecture and how it determines functional modules.

Acknowledgements

We thank Joshua Weitz for fruitful conversations. This work was supported by the Spanish Ministerio de Ciencia e Innovación grants BFU2009-06993 (SFE) and FIS2009-12365 (RVS), the James McDonnell Foundation (RVS), the Marcelino Botín Foundation (RVS), the John Templeton Foundation (SFE), and the Santa Fe Institute (RVS and SFE).

References

Barber, M. J. (2007). Modularity and community detection in bipartite networks. *Phys. Rev. E* 76:066102.

Collins, S. R., Miller, K. M., Maas, N. L., et al. (2007). Functional dissection of protein complexes involved in yeast chromosome biology using a genetic interaction map. *Nature* 446:806-810.

Costanzo, M., Baryshnikova, A., Bellay, J., et al. (2010). The genetic landscape of a cell. *Science* 327:425-431.

Edlund, J. A. and Adami, C. (2004). Evolution of robustness in digital organisms. *Artif. Life* 10:167-179.

Elena, S. F., Solé, R.V., and Sardanyés, J. (2010). Simple genomes, complex interactions: epistasis in RNA virus. *Chaos* 20:026016.

Elena, S. F. and Sanjuán, R. (2008). The effect of genetic robustness on evolvability in digital organisms. *BMC Evol. Biol.* 8:284.

Elena, S. F., Wilke, C. O., Ofria, C., and Lenski, R. E. (2007). Effect of population size and mutation rate on the evolution of mutational robustness. *Evolution* 61:666-674.

He, X., Qian, W., Wang, Z., Li, Y. and Zhang, J. (2010). Prevalent positive epistasis in *Escherichia coli* and *Saccharomyces cerevisiae* metabolic networks. *Nat. Genet.* 42:272-276.

Kauffman, S. A. (1993). *Origins of order*. Oxford University Press, New York.

Lenski, R.E., Ofria, C., Collier, T. C. and Adami, C. (1999). Genome complexity, robustness and genetic interactions in digital organisms. *Nature* 400:661-664.

Macía, J., Solé, R. V. and Elena, S. F. (2011). The causes of epistasis in genetic networks. *Evolution* 66:586-596.

Misevic, D., Ofria, C. and Lenski, R.E. (2006). Sexual reproduction reshapes the genetic architecture of digital organisms. *Proc. R. Soc. B* 273:457-464.

Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Phys. Rev. E* 69:026113.

Ofria, C. and Wilke, C. O. (2004). Avida: a software platform for research in computational evolutionary biology. *Artif. Life* 10:191-229.

Sanjuán, R. and Elena, S. F. (2006). Epistasis correlates to genomic complexity. *Proc. Natl. Acad. Sci. USA* 103:14402-14405.

Segrè, D., DeLuna, A., Church, G. M. and Kishony, R. (2005). Modular epistasis in yeast metabolism. *Nat. Genet.* 37:77-83.

Van Driessche, N., Demsar, J., Booth, E. O., Hill, P., Juvan, P. Zupan, B., Kuspa, A. and Shaulsky, G. (2005). Epistasis analysis with global transcriptional phenotypes. *Nat. Genet.* 37:471-477.

Xu, L., Jiang, H., Chen, H. and Gu, Z. (2011). Genetic architecture of growth traits revealed by global epistatic interactions. *Genome Biol. Evol.* 3:909-914.

Zheng, J., Benschop, J. J., Shales, M., Kemmeren, P., Greenblatt, J., Cagney, G., Holstege, F., Li, H. and Krogan, N. J. (2010). Epistatic relationships reveal the functional organization of yeast transcription factors. *Mol. Syst. Bio.* 6:420.

Evolutionary Potential is Maximized at Intermediate Diversity Levels

Bess L. Walker and Charles Ofria

Department of Computer Science and Engineering

BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, MI 48823

{blwalker, ofria}@msu.edu

Abstract

Diversity in a population is often cited as a major facilitator for the evolution of new complex features. The intuition behind this dynamic is that if a population is exploring multiple regions of a fitness landscape, more opportunities exist to find new functionality. We use the digital evolution software platform Avida to explore the effect of multiple limited resources on phenotypic Shannon diversity and, in turn, on evolvability of populations. We show that Shannon diversity peaks at intermediate levels of resource availability to the population, and we map the evolvability of a complex computational task on this availability-diversity gradient. While the evolvability of the complex task is highest at intermediate availabilities, it does not peak at the same resource inflow level as Shannon diversity, and it is more robust than diversity in its response to inflow level. These results indicate that while phenotypic Shannon diversity may play into the evolution of complex features, the selective pressures caused by diversity cannot be the only — or indeed even the main — pressures behind such evolution.

Introduction

Resource inflow and availability is a major factor affecting ecosystem diversity (Tilman, 1982; Chesson, 2000; Hall and Colegrave, 2007; Abrams et al., 2008; Cardinale et al., 2009). Diversity, in its turn, has been shown by the evolutionary computation community to encourage the evolution of solutions to complex problems through a more thorough exploration of the fitness landscape (Friedrich et al., 2009). Here, we explore the effect of the availability of multiple limited resources on phenotypic Shannon diversity, and use this availability-mediated diversity gradient to examine the relationship between Shannon diversity and the evolution of complex features.

Of the many types and measures of diversity, we choose to examine phenotypic Shannon diversity. We choose phenotypic over genotypic diversity because, of the two, phenotypic diversity is most easily manipulated with limited resources. We would also expect different drivers of genotypic diversity to have radically different results depending on whether different genotypes form a cloud in one area of the fitness landscape or are spread widely. Although similar issues can exist with phenotypic diversity, the range

of interesting phenotypes in these experiments is far more constrained than the range of interesting genotypes. Phenotypic diversity therefore provides a more fair treatment. We choose to measure phenotypic diversity as the Shannon entropy of the phenotypes in the population because Shannon entropy effectively balances the two main interesting qualities in diversity: the range of possible results and the evenness in their distribution.

Lenski et al. (2003) have investigated the evolutionary origin of complex features using Avida, using Boolean EQU as the specific complex task under study. This is the most complex of the one- and two-input Boolean operations to calculate, requiring at least five logical NAND operations. An Avidian organism requires at least 19 coordinated instructions to perform EQU, including at least five *nand* instructions. The ancestor starts out with none of these instructions in its genome; Lenski et al. found that in the 23 of 50 populations that evolved EQU in their experiments, EQU evolved in anywhere from 51 to 721 mutational steps.

In practice, the evolution of EQU is dependent on rewarding building blocks: the one- and two-input Boolean tasks of lower complexity. When Lenski et al. evolved populations in environments where only EQU was rewarded, none of the populations evolved EQU. However, they also found that the evolution of EQU does not depend on any particular building block or pair of building blocks. In fact, EQU can evolve in many different ways and is not dependent on any one thing; all 23 of Lenski et al.'s EQU-evolving populations evolved building blocks in different orders and organized them differently in their genomes.

Methods

Study System

We use the digital evolution software Avida (Ofria and Wilke, 2004), allowing precise manipulation of resource availability and a complete record of the course of evolution. The Avida system consists of a grid of digital organisms, each with a simple circular genome composed of instructions from an assembly-like Turing-complete instruction set. Time in Avida is measured in *updates*; each update

corresponds to an average number of 30 instruction executions per organism in the population. Organisms running quickly will execute more than 30 instructions per update, while slow organisms will execute fewer.

By executing its genome, each organism is capable of self-reproduction; during this process, copy mutations may be introduced into the offspring's genome. Because the genetic instructions are drawn from a Turing-complete language, the organisms are also theoretically capable of any other Turing-computable task. The organisms have access to integers that they can manipulate; the researcher can choose to reward certain manipulations with additional CPU cycles. These additional CPU cycles allow the organism to execute its genome more quickly and thus increase fitness.

Avida also supports a resource system, allowing task rewards to be tied to these resources. We accomplish resource manipulation in this system by manipulating the resource supply rate. Of course, precise manipulation of resource supply rate is possible in laboratory chemostat systems, but the use of a digital system allows us to know every detail of the population at any point in evolution, and to achieve very high generation counts over the course of just a few hours for each replicate population. Complete information about the population allows a precise calculation of diversity, which in this asexual system we define as the Shannon entropy of expressed resource-use phenotypes. It also allows a concrete definition of the complex feature we are examining; in this case, the Boolean EQU operation (Table 1).

Function name	Boolean operation	Reward
NOT	$\neg A; \neg B$	$\times 2^1$
NAND	$\neg(A \wedge B)$	$\times 2^1$
AND	$A \wedge B$	$\times 2^2$
ORN	$(A \vee \neg B); (\neg A \vee B)$	$\times 2^2$
OR	$A \vee B$	$\times 2^3$
ANDN	$(A \wedge \neg B); (\neg A \wedge B)$	$\times 2^3$
NOR	$\neg A \wedge \neg B$	$\times 2^4$
XOR	$(A \wedge \neg B) \vee (\neg A \wedge B)$	$\times 2^4$
EQU	$(A \wedge B) \vee (\neg A \wedge \neg B)$	$\times 2^5$

Table 1: NAND-count-based task rewards in Lenski et al. The symbol “ \neg ” denotes negation, while semicolons separate symmetrical functions. An organism which performs a task has its current execution rate multiplied by the amount of the task's reward. Note that the EQU operation is sometimes known as XNOR.

Our experiments use a development version of Avida 2.12.3 with the default instruction set (inst-heads.cfg). The executable was compiled from publicly-available source code (at avida.devosoft.org); the specific git revision identifier of the code is e5ba9511df000bae780c8524abb6bd01987190a5.

We set the per-site copy mutation rate to .0025, while we

left the per-reproduction rates of insertion and deletion mutations at the Avida default value, .05.

The population structure is spatial; organisms reproduce into any of the nine cells surrounding and including the organism itself, preferring empty cells. The resource structure is non-spatial; all organisms access the same resource pools. Our world is a 60 x 60 toroidal grid, initially seeded with 3600 clones of an asexual ancestor organism capable only of reproduction. This ancestor is a modification of the default ancestor that ships with Avida, default-heads.org, to reduce its genotype from length 100 to length 50 by removing 50 lines of “blank tape” no-op instructions. Since the population experiences no bottlenecks, the entire world grid is populated throughout the experiments.

We used SciPy 0.10.1 to calculate statistics, and Matplotlib 1.1.0 to create graphs.

Configurations from Previous Experiments

In their investigation of the evolutionary origin of complex features, Lenski et al. rewarded digital organisms once for each distinct Boolean task performed. The value of each task corresponded to its complexity as approximated by the minimum number of Boolean NAND operations necessary for its performance (see Table 1).

Chow et al. (2004) investigated the relationship between resource inflow and diversity in Avida. They measured diversity as species richness; as the digital organisms are asexual, Chow et al. used a clustering algorithm based on phylogenetic distance to determine which genotypes belonged to the same “species”. Species richness in this system was the result of negative frequency-dependent selection due to multiple depletable resource pools. R_{INFLOW} units of resource flow into each resource pool at a constant rate over each update, and a percentage of each pool flows out, modeling a chemostat.

$$Inflow : R_{TASK} = R_{TASK} + R_{INFLOW} \quad (1)$$

$$Outflow : R_{TASK} = 0.01 * R_{TASK} \quad (2)$$

Chow et al. used the same set of Boolean computational tasks as Lenski et al., but linked each task to a separate resource pool. The amount of resource in a resource pool (R_{TASK}) determines the value of performing the associated task; the NAND-count is not considered. An individual organism depletes A_{TASK} units of resource from the task-linked pool when performing a Boolean task. This depletion results in negative frequency-dependent selection (Cooper and Ofria, 2002). Rewarding an organism for the performance of a task again consists of multiplying its current execution count by the amount of the reward.

$$A_{TASK} = 0.0025 * R_{TASK} \quad (3)$$

$$Depletion : R_{TASK} = R_{TASK} - A_{TASK} \quad (4)$$

$$Reward : \times 2^{A_{TASK}} \quad (5)$$

Limited-Resource Environment

Because the Lenski et al. environment determines task rewards purely by task complexity, it can be thought of as an environment with infinite resource inflow. Without the negative frequency-dependence of Chow et al.'s environments, populations converge to a single generalist genotype that performs all tasks. In the environments of Chow et al., high inflow rates result in populations that converge on a single genotype specialized on replication efficiency; these rarely perform more than one or two of the simpler Boolean tasks. This is because Chow et al. do not incorporate the difficulty of the task into the task's reward; at high resource abundance, there is little to no pressure to seek new resources, and thus no reason to do difficult tasks.

In studying the effect of resource supply on both phenotypic Shannon diversity *and* the evolution of complex features, it is useful to create environments in which both the difficulty of the task and its rarity in the population (via the availability of its associated resource) contribute to the reward an organism receives for performing that task. To that end, we have devised a limited-resource environment starting with Lenski et al.'s reward scheme, but where a linked resource pool mediates the amount of the reward as in Chow et al.; Table 2 describes this hybrid reward scheme.

Function name	# NAND	Depletion	Reward
NOT	1	A_{NOT}	$\times 2^{1 \cdot A_{NOT}}$
NAND	1	A_{NAND}	$\times 2^{1 \cdot A_{NAND}}$
AND	2	A_{AND}	$\times 2^{2 \cdot A_{AND}}$
ORN	2	A_{ORN}	$\times 2^{2 \cdot A_{ORN}}$
OR	3	A_{OR}	$\times 2^{3 \cdot A_{OR}}$
ANDN	3	A_{ANDN}	$\times 2^{3 \cdot A_{ANDN}}$
NOR	4	A_{NOR}	$\times 2^{4 \cdot A_{NOR}}$
XOR	4	A_{XOR}	$\times 2^{4 \cdot A_{XOR}}$
EQU	5	A_{EQU}	$\times 2^{5 \cdot A_{EQU}}$

Table 2: Hybrid task rewards, based both on task complexity and resource availability (A_{TASK} denotes the number of resource units an organism uses from the *TASK*'s pool). An organism that performs a task has its current execution rate multiplied by the amount of the task's reward.

Results and Discussion

Diversity Peaks at Intermediate Productivity

Of the inflow rates we examined, the intermediate R_{INFLOW} of 10 (Figure 1) had the highest diversity; observing the highly unimodal trend of this data, we conclude that diversity in this system peaks somewhere between an R_{INFLOW} of 3 and 30. At lower inflow rates, Shannon diversity drops off quickly; with too-low resource levels, each pool supports too few organisms to make any substantial impact on diversity. The number of phenotypes may remain

high, but the Shannon entropy of the population as a whole is low. At higher inflow rates, diversity drops more slowly as resources become so plentiful they might as well be unlimited. Indeed, at inflow levels of 1000 and above, negative frequency-dependent pressures are effectively removed. This result corresponds to the results in other studies of the effects of resource supply on diversity (e.g. Kassen et al., 2000; Chow et al., 2004; Hall and Colegrave, 2007)

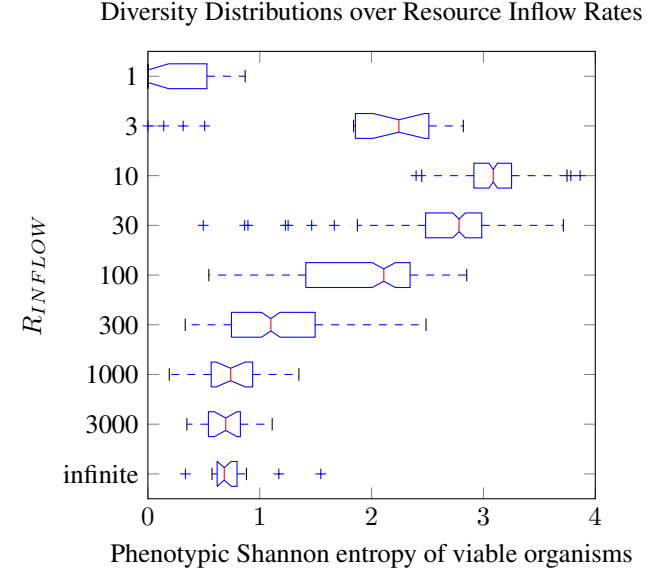


Figure 1: Diversity distributions across inflow rates, measured as the phenotypic Shannon entropy of all viable organisms in the population. Data for inflows 1, 3, 1000, 3000, and infinite are drawn from 20 populations; inflows 10, 30, 100, and 300 from 200 populations.

The Evolution of EQU is Common in Intermediate Productivities

We are now equipped to examine the evolvability of complex features on this resource inflow gradient, and to observe how it relates to the corresponding Shannon diversity gradient. In this case, we measure the evolvability of complex features by the proportion of populations that have evolved EQU by the end of 100,000 updates. At 20 populations per treatment (Figure 2), it is clear that the evolvability of EQU is highest at intermediate productivities. Indeed, between the intermediate inflow levels of 10 and 300 units per resource per update, the evolvability of EQU seems robust to increasing resource supply and decreasing phenotypic Shannon diversity.

To determine whether only complex tasks are sensitive to resource supply levels, we also examined the evolvability of the other 8 tasks rewarded in this environment (Figure 3). As a general trend, these tasks indicate that more complex

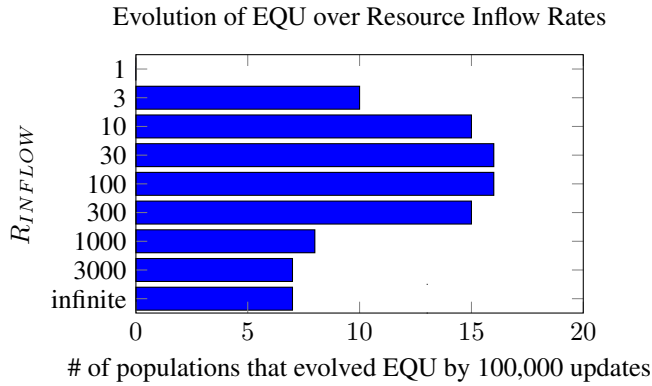


Figure 2: Evolvability of EQU across inflow rates, measured as the number of populations for which some genotype in the final population can perform EQU. Data for all inflows is drawn from 20 populations.

tasks are more sensitive to the resource supply level. Further investigation on this added axis of task complexity was beyond the scope of this paper.

We focused on the inflow rates in which populations were most successful in evolving EQU (10, 30, 100, and 300), and performed 10 times as many experimental runs at each to gain a higher resolution (Table 3). At this resolution, we saw that the evolvability of EQU is not truly unaffected by the variation of resource supply and Shannon diversity in this inflow range. The number of populations evolving EQU by the end of 100,000 updates is significantly higher at the 100 unit inflow rate than at the 10, 30, or 300 unit inflow rates. While these data do not indicate the precise R_{INFLOW} at which the evolutionary potential of EQU peaks, it is clearly a different — and greater — R_{INFLOW} than that at which phenotypic diversity reaches its peak.

R_{INFLOW}	10	30	100	300
#pops/200	141	152	171	152
p-value	<0.00001	<0.045	N/A	<0.045

Table 3: Number of populations out of 200 that evolved EQU at intermediate inflow rates. We performed a chi-squared test to determine if the evolvability of EQU for at least one inflow rate differed significantly from the rest ($p < .005$, $\chi^2 = 13.156$, 3 degrees of freedom). With this confirmed, we calculated the significance of each ratio’s difference from 171/200 with Fisher’s exact test, two-tailed, and corrected with the sequential Bonferroni correction; the $n=2$ correction was applied to both the $R_{INFLOW} = 30$ and $R_{INFLOW} = 300$ data, since they can be ordered arbitrarily.

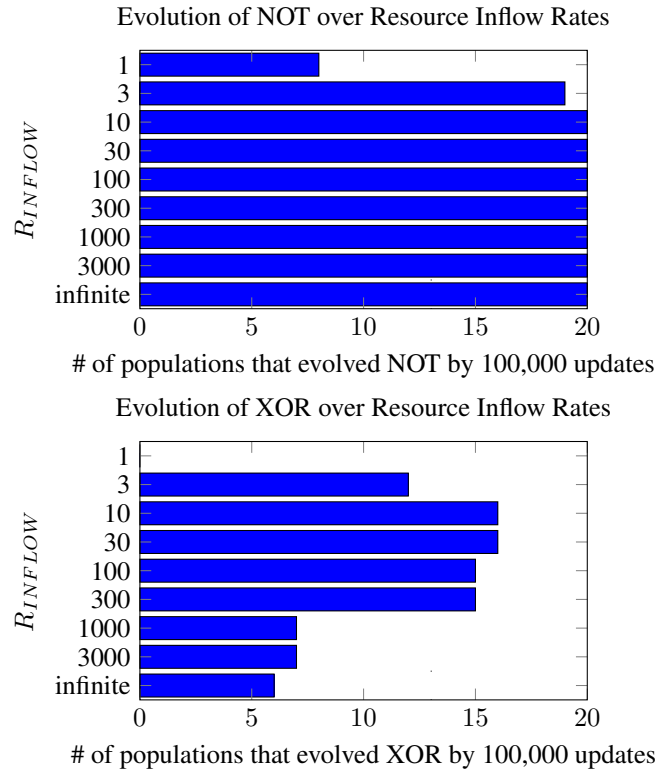


Figure 3: Evolvability of tasks requiring fewer NAND operations than the EQU task, measured as the number of populations for which some genotype in the final population can perform the task of interest. The first seven tasks (only NOT is shown) showed evolvability across resource inflows that was qualitatively similar to the NOT task shown here, with low evolvability at $R_{INFLOW} = 1$ and very high evolvability for all other inflow rates. The XOR task shows evolvability results qualitatively similar to the EQU task, with XOR being more evolvable at intermediate values of R_{INFLOW} . Data for all inflows and tasks is drawn from 20 populations.

Conclusions

We have seen that, for the inflow rates we tested, the phenotypic Shannon diversity of populations is highest at the 10 unit inflow rate (so likely peaks between R_{INFLOW} of 3 and 30). On the other hand, for the same set of inflow rates, the evolvability of this complex feature is highest at the 100 unit inflow rate (so likely peaks between R_{INFLOW} of 30 and 300). These ranges do not overlap; this difference indicates that diversity cannot be the only driver of the evolution of complex features, which is not unexpected. While the evolvability of complex features is indeed high at peak Shannon diversity, it seems that complex features may require more productive environments to evolve most often. We speculate that this greater resource availability and lesser

phenotypic diversity represent environments where more-abundant resources allow the desperate scramble for survival to relax slightly, allowing organisms to accumulate a collection of building blocks necessary for complex tasks.

These results indicate that evolutionary theory still has a great deal of work to do in tracking down the pressures responsible for the evolution of complex features. However, we have seen in this paper that the evolution of complex features is relatively robust, suggesting that the search for such pressures will not be akin to seeking a needle in a haystack — complex features evolve at a high rate at a large range of diversities in these experiments, and the number of times that EQU successfully evolved displays a decidedly unimodal nature. It is therefore likely to be similarly easy to track down the point of peak evolvability of complex features for other hypothesized pressures.

Future Work

In this paper, we have investigated only phenotypic Shannon diversity as caused by resource-based negative frequency-dependent selection. Negative frequency-dependent selection allows adaptive radiation in the Avida system's homogeneous environment, but it is not the only driver of diversity in nature. The relationship between the evolvability of complex features and diversity as driven by other factors (e.g. spatial structure, heterogeneous environments, or parasite pressures) certainly deserves investigation. Other measures of diversity ought also to be considered. Further, examination of the relationship between diversity and the evolvability of complex features only begins to explore the possible pressures driving the evolution of complex features. Although the mechanisms allowing complex features to evolve have been the subject of much investigation and debate (see Gregory, 2008, for an excellent overview), the exploration of pressures involved in the evolution of complex features has only begun.

Acknowledgements

We would like to thank Richard Lenski for helpful discussions about the relationship between diversity and the evolution of complex features, Heather Goldsby for her comments on the introductory section of this paper, and Michael Wiser for his comments and careful copyediting of the final draft.

This material is based in part upon work supported by the National Science Foundation under Grant No. CCF-0643952 and Cooperative Agreement No. DBI-0939454. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

- Abrams, P. A., Rueffler, C., and Kim, G. (2008). Determinants of the strength of disruptive and/or divergent selection arising from resource competition. *Evolution*, 62(7):1571–1586.
- Cardinale, B. J., Hillebrand, H., Harpole, W. S., Gross, K., and Ptacnik, R. (2009). Separating the influence of resource ‘availability’ from resource ‘imbalance’ on productivity–diversity relationships. *Ecology Letters*, 12(6):475–487.
- Chesson, P. (2000). Mechanisms of maintenance of species diversity. *Annual Review of Ecology and Systematics*, 31:343–366.
- Chow, S. S., Claus, W. O., Ofria, C., Lenski, R. E., and Adami, C. (2004). Adaptive radiation from resource competition in digital organisms. *Science*, 305(5680):84–86.
- Cooper, T. F. and Ofria, C. (2002). Evolution of stable ecosystems in populations of digital organisms. In *Proceedings of VIII International Conference on Artificial Life*, pages 227–232.
- Friedrich, T., Oliveto, P. S., Sudholt, D., and Witt, C. (2009). Analysis of diversity-preserving mechanisms for global exploration. *Evolutionary Computation*, 2003. CEC’03. *The 2003 Congress on*, 17(4):455–476.
- Gregory, T. R. (2008). The evolution of complex organs. *Evolution: Education and Outreach*, 1:358–389.
- Hall, A. R. and Colegrave, N. (2007). How does resource supply affect evolutionary diversification? *Proceedings of the Royal Society B: Biological Sciences*, 274(1606):73–78.
- Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95.
- Jones, E., Oliphant, T., Peterson, P., et al. (2001–). SciPy: Open source scientific tools for Python.
- Kassen, R., Buckling, A., Bell, G., and Rainey, P. B. (2000). Diversity peaks at intermediate productivity in a laboratory microcosm. *Nature*, 406(6795):508–512.
- Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423(6936):139–144.
- Ofria, C. and Wilke, C. O. (2004). Avida: A software platform for research in computational evolutionary biology. *Artificial Life*, 10(2):191–229.
- Tilman, D. (1982). *Resource Competition and Community Structure*. Princeton University Press, Princeton, NJ.

Is Evolution by Natural Selection the Algorithm of Biological Evolution?

Richard A. Watson

Natural Systems Group, University of Southampton, U.K.
raw@ecs.soton.ac.uk

Abstract

It is tempting to be confident that we know how biological evolution works. After all, we know a mechanism capable of producing adaptation, and we understand the necessary and sufficient conditions for this to occur, and those conditions are met in natural populations – the rest is surely just details. However, there can be many different algorithms that utilise a given underlying mechanism (sub-algorithm), and in other contexts we cannot assert that we know what algorithm is operating just because we identify a sub-algorithm it contains. Using sorting algorithms based on the mechanism of ‘compare and swap’ (as an analogue of evolutionary algorithms based on natural selection) we discuss three substantial ways in which an algorithm can be based on, and depend on, a mechanism and yet not be that mechanism, each of which has some bearing on natural processes of evolution: 1) unstructured versus structured applications of a mechanism, 2) data-independent versus data-dependent, 3) iterative versus recursive. In the context of computational algorithms more generally, it is easy to see that each of these issues corresponds to different algorithmic classes. We suggest that in natural evolution, it is not obvious that none of these issues apply, nor that the empirical evidence supports the view that an unstructured, data-independent and iterative interpretation of natural selection is sufficient to create biological evolution.

Biological Evolution and Natural Selection

Here we are interested in the adaptive aspects of evolution and the algorithmic principles that produce such adaptation. To begin, we must distinguish adaptive aspects of *biological evolution*, i.e., the phenomenon of adaptation that actually happens in the natural world (by whatever mechanism), from *evolution by natural selection*, ENS, i.e., the specific algorithmic account of biological evolution originating with Darwin. Of course, Darwin did not know a lot of the mechanical details that were filled-in later by neo-Darwinism, the Modern Synthesis and subsequent work. But here we are interested in the general form of the algorithm described by, for example, Lewontin (1970) – i.e. heritable variation in reproductive success. ENS, as used here, therefore refers to the standard model: i.e. an algorithm involving a population of individuals, reproducing at variable rates as determined by heritable characteristics that are susceptible to random variation. These basics have not changed since Darwin and

continue to form the basis of all working models for adaptation in evolutionary biology despite the many complexities of real biology that have become evident since Darwin. To take just one example, although evidence for the neutral theory of evolution (Kimura, 1985) might change our interpretation of evolutionary processes, few would argue that it invalidates ENS as an explanation of adaptive change. Indeed, it almost seems impossible, at least to some, that any such detail could alter the fundamental algorithm of evolution.

We take it as given that biology instantiates ENS. That is, ENS occurs in biological evolution (there is no need to reiterate the evidence for this). However, we wish to separate the conclusion that ENS *occurs in* biological evolution from the conclusion that the algorithm of adaptive biological evolution *is* ENS. Ordinarily, the notions involved are difficult to separate – or at least, care is not taken to separate them. The statement ‘evolution is true’, for example, fails to separate the claims that biological evolution has occurred (species have changed adaptively over time), that ENS has occurred (as described by Darwin), and/or that ENS is the mechanism by which biological evolution has occurred. A lot of emphasis is placed on showing that a biological population instantiates ENS with the implicit assumption that, if we show that it does, then we have shown that we know how biological evolution works. Does that necessarily follow?

Crudely, the issue that we want to discuss is something very simple: that a physical system can (trivially) instantiate more than one algorithm simultaneously; in particular, that an algorithm *B* can contain another algorithm *A*, ($B \neq A$). For example, an algorithm for matrix multiplication contains an algorithm for addition but it is not an algorithm for addition. Thus, biological evolution may *contain* ENS (and it does), but it might not *be* ENS. Accepting this logical possibility immediately and directly leads to the conclusion that no amount of evidence for ENS in biological populations can enable us to conclude that we know the algorithm of biological evolution. Also, despite the fact that addition is simpler than matrix multiplication, addition is not a more parsimonious explanation of matrix multiplication because it is not sufficient for matrix multiplication. Likewise an argument of parsimony does not enable us to conclude that we know the algorithm of biological evolution unless we can show that ENS (alone) is sufficient for biological evolution.

Of course, previous work has discussed at length evidence for the sufficiency of ENS to produce the biological

adaptation we observe in nature, e.g., (Sober, 1984; Neander, 1995; Bedau, 2008). However, such debates are often hampered by the implicit assumption that ENS is effectively a synonym for a natural account of biological evolution (in contrast to a supernatural ‘account’). Because of this assumption, the assertion that there must be a natural algorithm responsible for biological evolution forces the conclusion that ENS is that algorithm – and the result of this circular reasoning leads to the conclusion that ENS must be sufficient for biological adaptation. An inability to see that ENS might not be the algorithm of biological evolution (even though biological evolution surely depends on it) makes it impossible to discuss properly the possibility that ENS might not be sufficient to produce the adaptation we observe¹.

Nonetheless, two *uninteresting* possibilities for how biological evolution might contain ENS but not be ENS might come to mind: a) there exists an altogether different algorithmic process, operating over and above ENS, such that the existence of ENS in biology is sort of a coincidence/ or even a red herring, b) that biological evolution is some small variant of ENS, ‘ENS plus some bells and whistles’, but really, the fundamental nature of the algorithm is still ENS. A constructive discussion will require a carefully considered middle ground that neither depends on fantastical hypothetical alternatives nor on splitting-hairs. The issues we discuss are not merely hypothetical – and, in fact, the existence of relevant features/mechanisms in biological evolution is not in question. The more difficult issue is whether such features are minor details or algorithmically substantive.

The main contribution of this paper thus concerns issues of algorithmic equivalence. We want to address features that change the fundamental nature of an algorithm – but we also want to show that such features can nonetheless be rather subtle. In fact, we restrict ourselves further to cases where *A* and *B* have a particular kind of relationship such that *B* does not merely contain a mechanism (sub-algorithm) *A*, but *B* is *based on A*. Meaning that *B* depends on *A* (*A* is essential for *B*), and in a sense *B* is just *A* arranged in a particular manner, and the difference between *A* and *B* might come ‘for free’ (without design). Using concrete examples from another domain, we then investigate whether it can be the case that even though *B* is based on *A* in this restricted sense, it can nonetheless be the case that, we do not know how *B* works just because we know how *A* works. If so, this potentially prevents us from concluding that we know how biological evolution works even if granted that it is based on ENS.

Natural Selection + $X \approx$ Natural Selection?

The conditions that facilitate the process of evolution by natural selection – i.e. heritable variation in reproductive

¹As scientists we must be careful that we do not fall back on the following argument: 1) either it’s ENS or it’s supernatural, 2) it is not supernatural, 3) therefore it’s ENS. Even though the logic of the argument is correct, the conclusion is false because the first clause is false. Despite the fact that ENS occurs in nature and is capable of producing adaptation, it is not the only logical possibility (even if we restrict ourselves to mechanisms *based on ENS*). The assumption that questioning the sufficiency of ENS implies a willingness to entertain supernatural ‘accounts’ is potentially highly damaging to scientific debate.

success – are common in natural populations. The action of natural selection can be observed in natural populations and under controlled conditions; and, it is evidently capable of producing adaptation. The fossil record and genomic data show that all living things are connected in a tree of incremental (phenotypic and genetic) changes as the theory predicts. There is therefore no doubt that natural selection occurs and that it is fundamental to evolutionary change.

Given these facts, the claim that we know how evolution works seems reasonable. Dawkins, for example, states that “What Darwin achieved was nothing less than a complete explanation for the complexity and diversity of all life” (Dawkins, 2008). Obviously, “complete” is an overstatement. Many details have been filled-in/added-on over the last 150 years – including neutral evolution (Kimura, 1985), kin selection (Hamilton, 1964), niche construction (Odling-Smee et al, 2003), epigenetic inheritance (Jablonka & Lamb, 1995), self-organisation (Kauffman, 1993), symbiogenesis (Margulis & Fester, 1991), exaptation (Gould & Lewontin, 1979), ‘evo-devo’ interactions (Sommer, 2009), lateral gene transfer (Doolittle & Bapteste, 2007), compositional evolution (Watson, 2006), etc. – not to mention the molecular basis of inheritance – and there will surely be more. Moreover, some authors argue that issues such as these have a fundamental bearing on the underlying algorithm of biological evolution (see also Pigliucci, 2007). But others disagree – the hyperbole of Dawkins aside, the sentiment is that all of these ‘add-ons’ are merely contingent implementation details compared to the fundamental mechanism that drives it, ENS.

Underlying this there is perhaps a belief that ENS is the only adaptive algorithm that could *possibly* occur spontaneously in a physical substrate (and that therefore all other details *must* be either derivatives of it or unimportant). This assumption must be dispensed with. After all, prior to Darwin, no one could imagine *any* adaptive algorithm that could possibly occur spontaneously in a physical substrate. But Darwin showed that there exists at least one algorithm in that class. Is ENS really so fundamental that it is impossible for another algorithm to exist that is not simply a derivative of it? Consider a trivial counter-example; the optimization algorithm simulated annealing (Kirkpatrick et al, 1983). This occurs spontaneously in physical systems – in hot lumps of metal and other crystals as they cool – which was the inspiration for the computational algorithm in the first place. But simulated annealing is not natural selection and not a derivative of it. One might object that this example is merely the result of a physical dynamical system just doing what it does naturally – but so is natural selection, of course – there is nothing ‘other worldly’ about ENS. One might also object that simulated annealing is arguably a less sophisticated (weaker) algorithm than ENS, we would agree. Nonetheless, not all natural algorithms are necessarily ENS or derivatives of it.

However, in this paper we deliberately restrict ourselves to algorithms that, like all of the expansions mentioned above, *contain* ENS, and to algorithms that are based on ENS in a fundamental manner such that ENS is essential for their operation – i.e., no adaptation would occur without the inclusion of ENS. This might appear to concede that we are

merely talking about ‘add-ons’ – additions or extensions that do not change the fundamental underlying algorithm. But it is exactly the validity of this conclusion that we wish to discuss.

Algorithmic equivalence and subtle ways an algorithm can be based on a given mechanism

In the context of algorithms in general, it is easy to confirm that algorithm *B* can contain another algorithm *A* and yet not be algorithm *A* (see the matrix multiplication example above, or, in turn, the relationship of matrix multiplication to some signal processing algorithm, for example). Given that *B* contains *A*, we could argue that *A* is not an *incorrect* description of *B* but merely *incomplete*. But this would clearly be disingenuous in some cases (e.g., the missing details between addition and matrix multiplication are clearly fundamental). Likewise, it follows that, *in principle*, many different algorithms could contain natural selection and yet not be natural selection in a fundamental sense. But in the abstract, this point has little biological relevance. We need to restrict ourselves to biologically relevant algorithmic variants. However, rather than, at the other extreme, allowing the specific biological details (neutral evolution, evo-devo etc.) to drive the discussion, in this paper we take a different route.

Instead we discuss some specific but canonical ways in which algorithms can belong to fundamentally different algorithmic classes – despite all being based on the same underlying mechanism – simply by applying that mechanism in different ways. Namely, by applying that mechanism in different structural arrangements, in dynamic arrangements and in recursive arrangements:

- 1) unstructured vs structured applications of a mechanism,
- 2) data-independent vs data-dependent applications,
- 3) iterative vs recursive applications of a mechanism.

To this end we discuss at some length a number of non-evolutionary algorithms; in particular, different types of sorting algorithms all based on the mechanism of ‘compare and swap’. In this context it is clear that many different algorithms can all be based on and dependent on the same underlying mechanism, and yet belong to fundamentally different algorithmic classes. This enables us to discuss the relevant conceptual issues in a domain that is uncontroversial. We suggest that in light of such analogies, some important issues in evolution that presently seem inseparable can be teased apart, enabling us to ask clearer questions about biological evolution, and make clearer claims about our knowledge of it. This approach does, of course, have the weakness that we do not address specific biological mechanisms in detail – one may simply conclude that although our point may be true for sorting, it is not true for evolution. But it is our contention that in the domain we actually care about, biological evolution, the relevant facts (the algorithmic principles and their adaptive consequences) are not known – making it impossible to discuss the relevant issues. In the meantime, we aim to open up the relevant conceptual space to identify the relevant questions.

Our aims in this paper are therefore to discuss concepts such as algorithmic equivalence, and algorithmic classes, and in particular the implications of this for how we understand

evolution by natural selection. This is discussed at a largely conceptual level using analogies with other algorithmic domains. In each of the following sections we discuss the relevant issues with respect to algorithms in general (where the conclusions are uncontroversial), and then indicate their potential relevance to biological evolution. We then recap the implications and draw some general conclusions with respect to what we know about natural evolution.

Unstructured and structured applications of a mechanism

We begin with the subtle ways that the *structural context* of a mechanism can result in different algorithms.

Unstructured and structured sorting algorithms

Consider algorithms for sorting a list of numbers. Many sorting algorithms can be described as multiple applications of a ‘compare and swap’ (C&S) operator:

Compare-and-swap($\langle a, b \rangle$):

If $a \geq b$ return $\langle a, b \rangle$ else return $\langle b, a \rangle$.

Bubble-sort, for example, iterates through a list repeatedly, applying C&S to adjacent numbers in the list. One can visualise the order in which C&S operations are applied as a *sorting network* (Fig. 1) (Knuth, 1973; Cormen et al, 2001). Finding sorting networks with minimal number of comparators (and/or minimum depth) for a given number of inputs is a favourite sport of computer science (including, e.g., Hillis’ (1990) use of coevolutionary methods). Many different sorting algorithms (with different time complexities) can be described as sorting networks (Fig. 1).

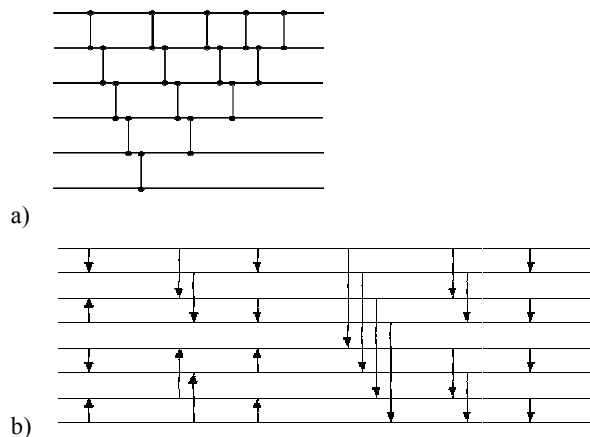


Figure 1: Sorting networks. Horizontal lines carry inputs that flow from left to right, vertical junctions apply C&S operations – a) Bubble-sort (the number of C&S operations can be reduced by 1 each iteration through the list as shown, giving a time complexity of $N(N-1)/2$ rather than N^2). b) Bitonic-sort, time complexity $\Theta(N \log(N)^2)$ (Batcher 1968).

Bubble-sort and Bitonic-sort both involve many C&S operations, and no sorting would occur without C&S; C&S is essential for sorting to occur. But even though the only difference between them is how the C&S operations are arranged, Bubble-sort and Bitonic-sort are not two different

descriptions of the same algorithm. This is evident in the observation that the two algorithms have different time complexities, and therefore, for a given time limit, one can sort lists of a given size correctly that the other cannot – i.e. Bubble sort does *not* sort correctly in time $N \cdot \log(N)^2$.

Note also that compare and swap itself is an algorithm. Both Bubble-sort and Bitonic-sort *contain* the C&S algorithm. But neither of them *are* C&S – there is more to them than that. It is therefore not the case that describing an algorithmic mechanism, *A*, that is essential for and contained in an algorithm, *B*, is the same as describing the algorithm *B* even if *B* is essentially just a particular arrangement of *A*. In these examples, the ‘structural context’ in which C&S occurs is necessary in order to describe Bubble-sort or Bitonic-sort. Note that it would be true to say that, given the right structural context, C&S results in sorting. But this still would not distinguish whether it was Bubble-sort or Bitonic-sort that had been implemented. Note also that some arrangements of C&S operations do not result in correct sorting for all inputs. Thus C&S is not sufficient for sorting.

Of course, Bubble-sort and Bitonic-sort are very special arrangements of C&S operations. However, a random arrangement of comparators with sufficiently many C&S operators would sort correctly so long as all comparators are pointing the same way – no other organisation is required. The expected time complexity (expected number of operators needed to sort correctly) for such a network is no more than N^2 times more than the time complexity of Bubble-sort, for example, (consider the probability of placing a required comparator between a particular pair of lines, and the fact that extra comparators do not hinder sorting). Such a *Random sort* network can reasonably be described merely as ‘lots of C&S operations’ since the structural context is minimal.

It is also instructive to consider the possibility of sorting networks that are structured in only subtle ways – e.g., such that nearby lines have a higher probability of a comparator being placed between them. If we restrict comparators to adjacent lines only, then the time complexity of *Adjacent-only random sort* is no more than N times the time complexity of Bubble-sort, for example – since Bubble-sort uses only this type of comparator (and the probability of placing a particular comparator is now $1/(N-1)$).

Unstructured and structured natural selection

We explore the analogy that ‘C&S is to sorting’ what ‘natural selection is to biological evolution’ (see Box 1). The sorting examples show that the structural context of an algorithm *A* (e.g., C&S) can change the algorithm *B* (containing *A*) in substantive ways. Thus even if biological evolution contains ENS we cannot necessarily conclude that the algorithm of biological evolution is ENS even if the only difference is how ENS is ‘arranged’. This would be reasonable only if (like Random sort) the structural context of natural selection in biological evolution was minimal – in this case biological evolution would be nothing more than ‘lots of natural selection’. But if biological evolution requires natural selection to be applied in a particular structural context then that could constitute a substantially different algorithm. This

To flesh-out the analogy, consider the ‘compare and copy’ (C&C) operator below:

Compare-and-copy($\langle a, b \rangle$):

If $a > b$ return $\langle a, a \rangle$ else return $\langle b, b \rangle$.

We can plug-in this operator in place of C&S into the above sorting algorithms. This will produce algorithms that take a list of N numbers as input and output a list of numbers that has multiple copies of numbers from the input in proportion to the number of two-player tournaments that they win. This is a simple selection algorithm or the reproduction part of an evolutionary process. We could likewise define a probabilistic version of this operator (copying a over b with probability that takes account of the ratio of their fitnesses) if that were desirable. A *compare-and-copy-with-variation* operator would provide all characteristics of heritable variation in reproductive success. To produce multi-generation evolution one would need to repeatedly call the sorter with the output of the previous ‘generation’. (See the *Microbial GA*, (Harvey, 2011), for a genetic algorithm using a steady state strategy with pairwise tournaments and in-situ variation (including sexual recombination) – but no structured context, by default).

Note that the time complexity of the sorting algorithms would then transform into the time complexity required to make N copies of the biggest number (there are easier ways to do that, but that’s not the point here). E.g., Adjacent-only random sort, given the C&C operator instead of C&S, would require less generations on average than Random sort to converge to an output where the biggest number is copied N times. Therefore, given limited time, Adjacent-only random sort could produce convergence in some cases where Random sort could not.

Box 1: From sorting algorithms to evolutionary algorithms

runs counter to the assumption that any algorithm based on natural selection is the same algorithm regardless of context.

In evolutionary theory it is well known that population structure changes the effective unit of selection. That is, *relatedness* in kin selection theory does not measure genetic relatedness in an absolute sense, but rather the genetic relatedness of the individuals that interact compared to the genetic relatedness of the population as a whole (Michod & Hamilton, 1980). Population structure therefore changes relatedness. Kin selection, or inclusive fitness theory, provides an explanation for differing levels of cooperation in a population, for example – i.e., different social outcomes. Multi-level selection theory (Wilson, 1992), including type-1 group selection, extends these principles. In principle, something as simple as the fact that a population is spatially embedded (altering who interacts with whom and who competes with whom) thus alters the structural context in which natural selection applies, e.g. by making proximal individuals more likely to participate in a competitive interaction than distal individuals (compare with Adjacent-only random sort). Then consider gene selection in the context of multi-cellular organisms and how much these ‘vehicles’ (Dawkins 1976) structure the context of genic selection. These

observations are not usually taken to imply a different algorithm for biological evolution. Should we conclude that these issues are minor modifiers on ENS? Or, that structured ENS, like structured sorting, constitutes a different algorithm? We suggest, it is not so obvious that structuring does not change the algorithm of evolution, nor that unstructured natural selection captures what is important about biological evolution any more than unstructured C&S captures sorting.

Data-independent and data-dependent applications of a mechanism

Clearly, the output of an algorithm is sensitive to its input. But the way in which an algorithm operates can also be sensitive to the input, and again, this can be rather subtle.

Data-independent and data-dependent sorting

In a sorting network, the sequence of C&S operations is fixed (part of the interest in them derives from the fact that their fixed arrangement makes them suitable for implementation in hardware). But other sorting algorithms exist that do not have this property. *Merge-sort*, for example, is a data-dependent algorithm. Merge sort depends on a Merge procedure (applied recursively) which takes two sorted lists (length $k/2$) and combines them into a single sorted list (length k). Each Merge requires only order N C&S operations. The recursive application of Merge, in effect, combines lists of length 1 (necessarily already sorted) into successively bigger lists until a complete list is returned. Quick-sort (which has an ‘in situ’ version, using no additional registers) is a sorting algorithm working on similar principles but ‘bottom-up’. Merge-sort and Quick-sort have (optimal) total time complexity $O(N \log N)$.

The Merge procedure compares the items from the tops of the two sorted input lists and transfers the smaller to the output list. This reveals a new top item in one of the lists. By repeating until the lists are empty, a fully-sorted output list is created. Note that there is no fixed order to the comparisons made in Merge – the n^{th} (for $n > 1$) comparison made depends on the outcome of the $(n-1)^{\text{th}}$ comparison and all previous comparisons. E.g. the first comparison is between A1 and B1 (the first elements of each list), then the second comparison is between A1 and B2, if B1 was greater than A1, whereas it is between A2 and B1, otherwise. Thus we cannot describe Merge-sort as any fixed ordering of compare and swap operations. Put another way, to implement Merge-sort in a sorting network would require a network where the result of a comparison at one point in the network influenced the presence or absence of a comparator down-stream.

It is instructive to consider a Random sorting network with a simple kind of data-dependence. For example, suppose that whenever a comparator does not result in a swap, nearby downstream comparators on the same lines are skipped. We can see that this might increase the efficiency of the sort by avoiding redundant comparators in some cases. More sophisticated local rules are also worth contemplating; e.g., if neither of the inputs to a comparator were altered since the last comparator on those lines, skip the comparator. Such a rule could be used in conjunction with a ‘fully-connected

sorting network’ – i.e. where all N^2 comparators are repeated N times. This network sorts correctly (Bubble sort is a subset of this network), and the data-dependence rule cannot prevent it from sorting correctly, and with the data-dependence rule it would use much less than the N^3 comparators present.

Note that even for a data-dependent algorithm there is a trace-back through time such that, at every point in time, we can explain a new list-ordering given the previous list-ordering and the application of the C&S operation applied at that point in time. But that is true for Merge-sort just like it is true for Bubble-sort or Random-sort – i.e., post hoc analysis shows that there is a sequence of C&S operations and, given that they occurred in that order, they explain the correct sorting. But the existence of such a trace (per se) does not distinguish which algorithm we are tracing or explain how they came to be in that order.

Data-independent & data-dependent natural selection

In the examples of contextual structuring we discussed above (e.g., population structure, kin selection, vehicles, multi-level selection) we assumed that these structures were constant or provided by extrinsic factors (e.g., spatial embedding or happenstance contingency). But, of course, they are also influenced by the action of natural selection itself. For example, the evolution of individual traits that affect habitat preference inevitably affect population structure and thus relatedness. Recent work (Powers, 2010; Powers et al, 2011; Snowdon et al, 2009) has begun to investigate the evolution of individual traits that affect the level of selection via *social niche construction* (Powers, 2010). This is a mechanism where (by analogy with *niche construction*, Odling-Smee et al, 2003) an organism alters its social context (who it interacts with and how much) and thereby affects the selective pressures on its social behaviour (e.g. cooperation). This fits directly with well-known theory relating population structure to social evolution (e.g. spatial or grouped population structures promote cooperation; Nowak & May, 1992). But whereas most studies assume that population structure is a given, social niche construction includes individual traits that alter population structure (e.g., via habitat preference, or selective adhesion, or the evolution of vertical transmission mechanisms). One particular study (Powers et al, 2011) investigates the evolution of initial group size in an aggregation and dispersal process and shows that individual natural selection drives group size down to increase cooperation (Szathmary, 2011). We have been investigating analogous mechanisms in various domains, in particular in adaptive networks (Gross & Sayama, 2009) where the topology of the network affects the behaviour on the network, and reflexively, the behaviour on the network affects the network topology (Watson et al, 2010; 2011a; 2011b).

Thus, by straight-forward means, the outcome of natural selection at one point in time can affect the way in which natural selection is applied at a future point in time (see also Neander, 1995). Thus, biological evolution is data-dependent. In principle, this puts it in a fundamentally different algorithmic class from data-independent natural selection. Of course, one might argue that Lewontin’s formulation, for

example, does not categorically exclude data-dependence (since the possibility is not mentioned). But the omission is potentially as substantial as saying that Merge-sort is lots of C&S without mentioning that the ordering depends on input.

Note also that there is a trace back through time such that, at every point, we can explain a new state of a population as a result of ENS acting on individual traits. But that does not distinguish which evolutionary algorithm we are tracing; in particular, whether the population structure that determined the structure of the trace was data-dependent or not. Thus, the fact that evolved organisms fit into a tree of life does not mean we can conclude that evolution is data-independent ENS (both data-independent and data-dependent ENS algorithms would have the property that results fit into such a tree).

Iterative versus recursive applications of a mechanism

Merge-sort, as well as being data-dependent, is also a recursive algorithm. The recursive application of a mechanism can result in a substantially different algorithm from iterative applications. Again, this has interesting analogues in biology.

Iterative versus recursive sorting

Bubble-sort is a simple iterative algorithm. Merge-sort (like Quick-sort) is a recursive algorithm. It sorts a list by dividing it in two, sorting each sub-list *using Merge-sort* (i.e., dividing it in two, sorting each sub-list *using Merge-sort*, and merging the sub-lists back together using the Merge procedure), and merging the sub-lists back together using the Merge procedure. To prove that Merge-sort sorts correctly we can use a proof by induction. First we show that a list of just one number is already sorted. Then we show that the *Merge* procedure, given two sorted input lists, produces one sorted list containing the numbers from both.

Note that the Merge procedure is not in itself a sorting algorithm – it will not produce sorted output from arbitrary inputs, only from two pre-sorted lists. Thus if we describe the Merge operation on its own, i.e. without the context of the recursive structure, it does not describe a process that sorts.

(For interest, it is not too hard to define a sub-network that carries out Merge using only order N C&S comparators, as Merge does, by starting with a full set of N^2 comparators and using data-dependence that turns off downstream comparators based on the outcome of upstream comparators, as mentioned previously. Thus, a Merge-sort network could be constructed using such dynamic sub-networks arranged appropriately).

Suppose we were to jump into a trace of the Merge-sort algorithm at a particular level of recursion; perhaps the last Merge before the sorted output is produced (i.e., two lists of $N/2$ into one list of N). We could explain (using data-dependent C&S operations) how Merge gets Merge-sort from this point in its operation to the final sorted output. However, this would fundamentally fail to explain the sorted result because it fails to explain how the two sub-lists came to be sorted at this stage of operation. We could try to explain this by saying that more Merging was involved; but note that it would not be more Merging at the same level of description.

We have to refer to Merging at multiple levels of organisation – i.e., no one level of Merge explains Merge-sort.

Iterative versus recursive natural selection

Recursion is obviously a very special algorithmic structure. But multiple nested levels of structural organisation are ubiquitous in nature – in both evolved and non-evolved systems (Lenaerts et al, 2005). The major transitions in evolution (Maynard Smith & Szathmari, 1995) describe just such a multi-scale structure, applying natural selection (in a structured and data-dependent manner) at many scales of organisation. Maynard Smith & Szathmari describe a set of transitions that have been fundamental in the evolution of complexity. These events, including for example the transition from self-replicating molecules to protocells and unicellular organisms to multi-cellular organisms, share the property that “entities that replicated independently before the transition can replicate only as part of a larger whole after the transition”. These processes are therefore entangled with issues such as changes in the unit of selection (Okasha, 2006; Michod, 1999; Buss 1987), and the ‘de-Darwinisation’ of lower level units and ‘Darwinisation’ of new higher-level units along various dimensions (Godfrey-Smith, 2009).

Note that jumping in at a particular level within this hierarchy to try and describe how natural selection proceeds at that one level of organisation would not describe the algorithm responsible for the evolutionary outcomes we observe because it would not explain where the inputs to this level of organisation came from. At one of the lower levels of organisation, this is loosely related to Sober’s (1984) position that natural selection can explain why a population exhibits trait *a* in preference to trait *b*, but not how either of those traits originated. It is a little too easy to simply assert that they originated from the prior action of ENS because we may be conflating different descriptive levels when we do this.

Put another way, consider the necessary and sufficient conditions for ENS described by Lewontin – heritable variation in reproductive success. Notice that all these terms require us to define the units we are talking about so that we can define reproduction (and Darwinian fitness/reproductive success), heritability and variation. For example, we could focus on the level of genes (as Dawkins advocates), then we can talk about the heritability of genes given a set of genetic variation operators (mutation and recombination), and selection on genes (either in the context of cells or sexual organisms), and given a physical substrate that defines how well a given genetic sequence survives and replicates. But clearly, a lot of machinery is already assumed here, and not all of it obviously comes ‘for free’ from the biophysical properties of molecules. Sexual recombination, for example, is an evolved mechanism that radically changes the effective unit of selection from genomes to genes (Watson, 2005) – without some mechanism that enables genes to be inherited individually the premise of genic selection is meaningless, and we would be talking only about genome selection.

The point about the Merge procedure is not merely that the inputs (sorted sub-lists) are variable in size or that sub-lists of different sizes are relevant at different stages of the process.

The point is that the Merge procedure (despite containing lots of C&S) is not a sorting algorithm at all, and only when one appreciates that Merge-sort is recursive, and therefore continually redefines the inputs to the Merge procedure, do we understand how Merge-sort produces sorted outputs. Likewise, the point is not merely that the terms of reference in biological evolution are a bit slippery – a bit difficult to define clearly (this point has been made many times, e.g. see Godfrey-Smith, 2009 for many interesting examples). The point is that there is not necessarily *any* one set of terms that satisfies the requirements of the process.

'Self-structuring' & Opaque Consequences of ENS

Recursion involves a process turned upon itself. More generally, the idea that evolution can modify its own operation is discussed in the evolution of evolvability (Kirchner & Gerhart, 1998; Sterelny 2011) and, e.g. evolved exploration distributions (Toussaint & von Seelen, 2007; Parter et al, 2008) that alter the space of phenotypic possibilities on the fly. Likewise, the evolution of new genetic mechanisms (non-random genetic variation mechanisms, e.g. via mobile genetic elements; Shapiro, 2011) can alter the space of genetic possibilities – analogous to 'self-modifying code' in computer science. In the major transitions we contemplate evolution modifying its own operational units.

Such recursive and self-referential notions of evolution present a concept of evolution that continually 'reinvents itself'; changing the level of selection, forming new mechanisms of heredity and creating new evolutionary units at successive scales of biological organisation. Thus evolutionary processes co-create the structural context of selection, effectively re-defining the evolutionary process (Sterelny, 2011; Calcott & Sterelny, 2011; Godfrey-Smith, 2011). We refer to this as '*self-structuring evolution*'.

The analogy suggests that explaining what is going on in some of these biological processes, especially in the major transitions and self-structuring evolution, is not captured by ENS, any more than Merge-sort, for example, is captured by unstructured, data-independent and iterative applications of the compare and swap mechanism. If this is the case, then *plain ENS*, i.e. unstructured, data-independent and iterative ENS, is not necessarily the algorithm responsible for producing biological adaptation even though it is evidently capable of producing some adaptation.

However, the question then becomes, is self-structuring ENS merely plain ENS given the right kind of substrate or materials? That is, *given appropriate conditions*, can plain ENS create structured, data-dependent and recursive ENS? If so, then any failure of plain ENS to explain biological evolution seems to be merely an epistemological issue – i.e., a failure to comprehend or deduce the opaque consequences of the original simple algorithm. We have some sympathy for this position. But plain ENS is not going to create structured ENS in all substrates/ environments – some substrates won't allow data-dependence or recursion for example. We would argue that understanding the conditions for plain ENS to become self-structuring is really a necessary part of describing

the algorithm (Bedau, 2008) – and such understanding is not captured by the plain ENS algorithm per se.

Moreover, the process that transforms plain ENS into structured ENS is not *necessarily* ENS itself. We might use the term 'self-organisation' to cover a multitude of possibilities with respect to order that comes 'for free' in physical systems (although note that here we are talking about self-organisation of the algorithm itself, not merely of the object/material that ENS operates on). We have been investigating a more specific mechanism of *associative induction* (Watson et al, 2010; 2011a; 2011b; submitted) that arises 'for free' in adaptive networks, for example. The hierarchical form of self-structuring evolution that results has a fundamentally different algorithmic capability from plain ENS (Mills, 2010; Watson et al, submitted). This implies that the algorithmic nature of biological evolution is not merely a point of view but can be settled empirically.

Conclusions

Understanding how compare-&-swap or Random sort works is a long way from understanding how Merge sort works. More generally, knowing that an algorithm sorts, and that it contains C&S, does not tell us whether that algorithm is an unstructured, data-independent and iterative algorithm (like Random sort) or, at the opposite extreme, a structured, data-dependent and recursive algorithm (like Merge sort). Thus, it is not the case that we necessarily know how an algorithm *B* works, even if we know that it contains a known algorithm *A*. Moreover, the examples of sorting algorithms show that *B* can belong to fundamentally different classes of algorithm even when the relationship between *A* and *B* is highly restricted such that *B* not only contains *A* but is *based on A*: Specifically, *B* depends on *A*, *A* is essential for the operation of *B*; *B* is, in a sense, just an arrangement of *A* (albeit perhaps a dynamic and/or recursive arrangement) and that arrangement can in some cases be built-up using only local restrictions and/or simple restructuring principles.

This shows that conditions that produce structured, data-dependent and/or recursive applications of a mechanism can result in an algorithm that is in a fundamentally different class from unstructured, data-independent and iterative applications of the same mechanism. Thus, even if we grant that biological evolution not only contains ENS but is *based on* ENS in this restrictive sense, no amount of evidence for the existence of ENS in nature enables us to conclude that we know the algorithm of biological evolution.

Parsimony would preclude the need to consider alternative algorithms for biological evolution if, but only if, it was shown that ENS was a sufficient algorithm to produce biological evolution. Thus, consider the statement: *There are no known examples of extent organisms or adaptations that could not plausibly have been produced by ENS given appropriate conditions/arrangements*. And compare with: *There are no known examples of sorting that could not plausibly have been produced by compare-and-swap given appropriate conditions/ arrangements*. Or for that matter: *There are no known examples of matrix multiplication that could not plausibly have been produced by addition given*

appropriate conditions/ arrangements. Obviously the latter statements can only be true because the ‘appropriate conditions’ clause can bury substantial algorithmic structure. The question is thus whether ‘the conditions’ of biological evolution conceal substantial algorithmic structure. We have discussed how these conditions might include structuring, data-dependence and recursion and that doing so would change the fundamental nature of the algorithm. We cannot therefore accept the first of these three statements as evidence that ENS is algorithmically sufficient for biological evolution.

In conclusion, we suggest that it is not at all clear that biological evolution is unstructured, data-independent and iterative – indeed, we have discussed specific evidence to the contrary. Thus, notwithstanding the fact that biology instantiates ENS, it is certainly not for granted (arguments of parsimony included) that evolution by natural selection is the algorithm of biological evolution.

Acknowledgements: Thanks to Chris Adami for motivating this exploration. Thanks to Hywel Williams, Paul Ryan, Jason Noble, Adam Davies and Miguel Gonzalez Canudas for discussion of the manuscript, and Keyvan mir Mohammad Sadeghi for background research on sorting networks.

- Batcher, K.E. (1968) Sorting Networks and their Applications. *Proc. AFIPS Spring Joint Comput. Conf.*, Vol. 32, 307-314.
- Bedau, M.A. (2008). The evolution of complexity. In Thomas Pradeu, et al., eds., *Mapping the Future of Biology: Evolving Concepts and Theories*.
- Buss, LW, (1987), *The Evolution of Individuality*, Princeton Press, NJ.
- Calcott, B. & Sterelny, K. (2011) "Introduction: A Dynamic View of Evolution" in *The Major Transitions in Evolution Revisited*, MIT Press.
- Cormen, T.H., C.E. Leiserson, R.L. Rivest, C. Stein (2001) *Introduction to Algorithms. 2nd edition*, MIT Press.
- Darwin, C. (1872) *The Origin of Species*, London, John Murray.
- Dawkins, R. (1976), *The Selfish Gene*, Oxford University Press.
- Dawkins, R. (2008) *The Genius of Charles Darwin, Part 1. Life, Darwin & Everything*. Channel 4 television (broadcast August 2008).
- Doolittle, W. Ford, and Eric Baptiste. (2007). Pattern pluralism and the tree of life hypothesis. *PNAS* 104(7): 2043–2049.
- Godfrey-Smith, P. (2009). *Darwinian Populations and Natural Selection*. Oxford.
- Godfrey-Smith, P. (2011) "Darwinian Populations and Transitions in Individuality" in *The Major Transitions in Evolution Revisited*, MIT Press. 65-82.
- Gould S.J. & Lewontin, R.C. (1979) The Spandrels of San Marco and the Panglossian Paradigm. *Procs. Roy. Soc. B* 205:1161, 581-598.
- Gross, T. & Sayama, H. (2009) *Adaptive Networks. Theory, Models and Applications*. Springer-Verlag: Berlin.
- Hamilton W. D. (1964). The Genetical Evolution of Social Behavior. *J. Theor. Biology* 7 1-16.
- Harvey, I. (2011) The Microbial Genetic Algorithm, *Procs. ECAL 2009*, 126-133.
- Hillis, W.D. (1990) "Co-evolving Parasites Improve Simulated Evolution as an Optimization Procedure" *Physica D: Nonlinear Phenomena*. 42(1-3) 228–234.
- Jablonka, E. & Lamb, MJ (1995) *Epigenetic Inheritance and Evolution: the Lamarckian Dimension*, Oxford University Press
- Kauffman S. (1993) *The origins of order: Self organization and selection in evolution*. Oxford.
- Kimura, M. (1985) *The neutral theory of molecular evolution*. Cambridge.
- Kirchner, M. & Gerhart, J. (1998). Evolvability. *PNAS*. 95:8420–8427
- Kirkpatrick, S., Gelatt, C.D. & Vecchi, M.P. (1983) Optimization by Simulated Annealing. *Science*. 220 (4598): 671–680.
- Knuth, D.E. (1973) *The Art of Computer Programming, Vol. 3 - Sorting and Searching*. Addison-Wesley.
- Lenaerts, T., Chu, D., Watson, R.A. (2005) Dynamical hierarchies, *Artificial Life*. 11(4):403-405
- Lewontin, R.C. (1970) "The Units of Selection," *Annual Review of Ecology and Systematics*, 1: 1-18.
- Margulis, L. & Fester, R. (1991) *Symbiosis as a Source of Evolutionary Innovation*. Cambridge, MA: MIT Press
- Maynard Smith, J. & Szathmary, E. (1995) *Major Transitions in Evolution*. W. H. Freeman.
- Michod, R. E. & Hamilton, W. D. (1980) Coefficients of relatedness in sociobiology. *Nature* 288, 694 – 697.
- Michod, R.E., (1999), *Darwinian Dynamics, Evolutionary Transitions in Fitness and Individuality*. Princeton Univ. Press
- Mills, R.M. (2010) *How Micro-Evolution Can Guide Macro-Evolution*, PhD thesis, ECS, Southampton.
- Neander, K. (1995) Pruning the Tree of Life, *British Jnl. for the Philosophy of Sci.* 46(1): 59-80.
- Nowak MA, & May, R.M. (1992) Evolutionary Games and Spatial Chaos, *Nature* 359, 826-829.
- Odling-Smee, F. J., Laland, K. N., and Feldman, M. W. (2003). *Niche construction: the neglected process in evolution*. Monographs in population biology; no. 37. Princeton University Press
- Okasha, S. (2006). *Evolution and the Levels of Selection*. Clarendon.
- Parter, M. et al (2008) Facilitated Variation: How Evolution Learns from Past Environments to Generalize to New Environments. *PLoS Comput Biol* 4(11).
- Pigliucci M (2007) Do we need an extended evolutionary synthesis? *Evolution* 61(12): 2743-2749.
- Powers, S.T. (2010) *Social Niche Construction*, PhD thesis, ECS, Southampton.
- Powers, S.T., Penn, A.S., Watson, R.A. (2011) The Concurrent Evolution of Cooperation and the Population Structures that Support it. *Evolution*. 65(6):1527–1543.
- Shapiro, J. (2011) *Evolution: A View from the 21st Century*. FT Press.
- Snowdon, J., Powers, S. & Watson, R.A. (2009) Moderate contact between sub-populations promotes evolved assortativity enabling group selection. *Procs. 10th Euro. Conf. on Artl. Life (ECAL 2009)*. 2:42-49.
- Sober, E.(1984) *The nature of selection: evolutionary theory in philosophical focus*. Chicago.
- Sommer, R.J. (2009). The future of evo-devo: model systems and evolutionary theory. *Nature Reviews Genetics* 10 (6): 416–422
- Sterelny, K. (2011) "Evolvability Reconsidered" in *The Major Transitions in Evolution Revisited*, MIT Press. pp: 83-100.
- Szathmary, E. (2011) To Group or Not to Group? *Science*. 334. 1648-9.
- Toussaint, M., & von Seelen, W. (2007) Complex adaptation and system structure, *BioSystems* 90: 769–782
- Watson, R.A. (2005) On the Unit of Selection in Sexual Populations. In *Advances in Artificial Life, (ECAL 2005)*.
- Watson, R.A. (2006) *Compositional Evolution: The impact of Sex, Symbiosis and Modularity on the Gradualist Framework of Evolution*. MIT Press.
- Watson, R.A., Buckley, C.L., Mills, R.M. (2010) Optimisation in 'Self-modelling' Complex Adaptive Systems. *Complexity*. 16(5):17-26.
- Watson, R.A., Jackson A., Palmius, N., Mills, R.M., Powers, S.T. (submitted) *The Evolution of Symbiotic Partnerships and Their Adaptive Consequences*.
- Watson, R.A., Mills, R.M., Buckley, C.L. (2011a) Global Adaptation in Networks of Selfish Components. *Artificial Life*. 17(3):147-66.
- Watson, R.A., Mills, R.M., Buckley, C.L. (2011b) Transforming the Scale of Dynamical Behaviour in Nearly Decomposable Systems, *Adaptive Behaviour* 19(4): 227-249.
- Wilson D.S. (1992). Complex interactions in meta-communities, with implications for biodiversity and higher levels of selection. *Ecology*, 73(6):1984-2000.

Coevolving parasites improve host evolutionary search on structured fitness landscapes

Hywel T. P. Williams

College of Life & Environmental Sciences, University of Exeter, Exeter, EX4 4PS, UK
h.t.p.williams@exeter.ac.uk

Abstract

Evidence suggests that host-parasite coevolution can often result in host diversification. However, the host traits that coevolve often have primary functions affecting growth, creating the potential for conflicting selection pressures. For example, bacteriophage often infect bacteria by binding to nutrient uptake receptors, thus diversification of bacteria due to coevolution with phage may have an impact on resource competition. This paper uses a model of bacteria and phage in a chemostat to study the impact of coevolution with phage on the evolution of host growth rates, when infection and growth are affected by the same trait. Comparing (co)evolutionary outcomes on different growth rate fitness landscapes, with and without phage, shows that coevolutionary diversification allows hosts to cross fitness valleys and improve search efficiency on rugged landscapes, although it also prevents the whole community from reaching global optima. In effect, coevolution with parasites increases *exploration* but decreases *exploitation* in host evolutionary search.

Introduction

All biological evolution is coevolution, in the sense that any evolving population has a selective environment formed of, and created by, other organisms. The biotic environment of an organism determines its direct ecological interactions and also ultimately shapes the character of its abiotic environment through niche construction effects (Odling-Smee et al., 2003; Williams and Lenton, 2008). In general, coevolution is a diffuse process, with most species having a negligible impact on the evolution of any focal species. However, when species interact closely and have a strong direct impact on each other's fitness, coevolution can be a significant determinant of evolutionary outcomes (Thompson, 2005). In host-parasite systems, for example, the parasite depends on the host for survival and reproduction. Such close interaction between hosts and parasites often leads to significant antagonistic coevolution, in which the host evolves to lessen the impact of the parasite, while the parasite evolves to maintain its infective and reproductive ability on the host (Buckling and Rainey, 2002; Woolhouse et al., 2002; Thompson, 2005).

A well-studied class of host-parasite interactions is the infection of bacteria by bacteriophage (viruses that infect bacteria). Phage are obligate intracellular parasites that infect bacteria by binding to a cell surface receptor before injecting their DNA. Phage are commonly classified as having either lysogenic or lytic lifestyles. Lysogeny involves the integration of phage DNA into the genome of the host cell, so that the phage is propagated by host reproduction for many generations until some trigger causes lysis (cell burst) and the release of new phage particles. In the lytic lifestyle, phage infect the host cell and subvert its metabolism to produce new phage, which are then released by lysis. Lytic infection suppresses host replication and always results in cell death, hence the interaction between bacteria and lytic phage is literally a matter of life and death; the phage must infect if they are to reproduce, while the cell must avoid infection if it is to survive. The selection pressure each partner exerts on the other is thus intense and antagonistic coevolution can produce rapid genetic change. For this reason, bacteria and lytic bacteriophage are often used for experimental studies of coevolution (Bohannan and Lenski, 2000; Buckling and Rainey, 2002; Brockhurst et al., 2007).

Coevolution has been hypothesised to have a significant impact on the diversity of hosts and viruses. Two models for the genetics of host-parasite coevolution are commonly discussed, the 'gene-for-gene' and 'matching-alleles' models (Agrawal and Lively, 2002). Each model makes a different prediction for diversity. The gene-for-gene model is adapted from plant-pathogen interactions and assumes that hosts have either resistant (*res*) or susceptible (*sus*) alleles at each infection locus, while at each paired locus the parasite has either virulent (*vir*) or avirulent (*avi*) alleles. At a single locus, infection occurs for cases: $\{res - vir, sus - vir, sus - avi\}$ but not for case $\{res - avi\}$. The gene-for-gene model predicts that coevolutionary arms races can occur, in which the host gains *res* alleles and the parasite gains *vir* alleles, but also predicts low-diversity outcomes in which the host and parasite populations are dominated by the most resistant and most infectious genotypes respectively. The matching alleles model is derived from self/non-

self recognition mechanisms in invertebrates and assumes that some form of genetic match between host and parasite is needed at relevant loci for infection to occur. Thus with matching-alleles genetics, a single host mutation can make infection impossible, but can be countered by a single parasite mutation. The matching-alleles model predicts diversification of hosts due to negative density-dependent selection from parasites; hosts diversify in order to escape infection, while parasites diversify as they counter-adapt. Various studies have demonstrated that stable polymorphisms are a common outcome from matching-alleles coevolution (Agrawal and Lively, 2002). Empirical data to support either matching-alleles or gene-for-gene as a general model for coevolution of bacteria with bacteriophage is equivocal. Other genetic systems have been hypothesised (Fenton et al., 2009; Hall et al., 2011) and it also seems likely that multiple mechanisms may operate concurrently (Agrawal and Lively, 2003; Fenton et al., 2012), so no simple generalisation can be made.

In natural microbial communities, coevolution with phage has been hypothesised as a possible explanation for widespread observations of high marine prokaryote diversity. Theoretical models of planktonic food-web ecology predict that selective viral predation can maintain host diversity by preventing dominance of host types that would otherwise monopolise available resources (the ‘kill-the-winner’ model (Thingstad and Lignell, 1997; Thingstad, 2000)). However, the action of coevolution on marine microbial communities is difficult to measure directly. Metagenomic data for marine prokaryotes so far suggests that phage are responsible for a high proportion of prokaryote diversity. The ‘constant diversity’ hypothesis (Rodriguez-Valera et al., 2009) states that bacteriophage maintain high diversity in prokaryote communities via negative density-dependent selection, based on evidence that high-diversity genomic islands in prokaryote genomes often code for traits associated with phage infection, e.g. surface proteins, CRISPR arrays, etc. The relationship between phage predation and host diversity was directly tested for *Prochlorococcus* and cyanophage by experiments that showed resistance mutations mostly occurred in hyper-variable regions of the genome and that any single phage strain could only infect a subset of the bacterial population (Avrani et al., 2011). Experiments with *Synechococcus* and cyanophage also showed rapid coevolutionary diversification of both host and phage (Marston et al., 2012).

Experimental coevolution with bacteria and bacteriophage in chemostats has shown that resistant mutant strains can enter the host population via selective sweeps, displacing susceptible strains; limited diversity can be sustained when trade-offs between growth rate and resistance allow stable coexistence of a growth-specialist and a resistance-specialist (Bohannan and Lenski, 2000). Meanwhile, experimental coevolution with bacteria and phage in batch culture

has repeatedly shown that hosts can rapidly acquire resistance to new phage while maintaining resistance to ancestral phage (Buckling and Rainey, 2002; Brockhurst et al., 2007). These ‘arms race’ dynamics are consistent with the gene-for-gene model and might be expected to lead to low diversity.

While it is hard to generalise the relationship between host-virus coevolution and diversity, there appears to be strong evidence that in many cases coevolution with viruses leads to host diversification (Rodriguez-Valera et al., 2009; Avrani et al., 2011; Marston et al., 2012). How does this virus-driven diversity affect the evolution of non-virus-associated traits? When coevolving traits do not affect other functions, coevolution and evolution may be orthogonal and proceed independently. However, coevolving traits often have a large impact on growth and/or reproduction (Lennon et al., 2007). Pleiotropic interactions may then lead to conflicts and trade-offs that shape adaptive trajectories. For example, phage often bind to nutrient uptake receptors on the bacterial cell surface, thus the evolution of these receptors is subject to (potentially conflicting) selection pressures for uptake efficiency and phage resistance (Rodriguez-Valera et al., 2009). Receptor mutations will thus affect both infection rate and resource competition, and may have opposing effects on each component of overall fitness.

The general scientific question addressed by this paper is whether diversification caused by coevolution with parasites has an impact on host adaptation to environmental selection pressures. Diversity is a pre-requisite for any form of evolution, since phenotypic differences form the basis of selectable variation in fitness. Hence a reasonable hypothesis is that increased diversity caused by coevolution with parasites (under a matching-alleles-like model) might lead to improved evolutionary search and faster adaptation of hosts. In particular, this study focuses on the evolution of a host trait that affects both growth rate and parasite infection, inspired by (amongst others) the natural example of bacterial resource uptake receptors that are also the attachment site for bacteriophage. A simple model of bacterial hosts coevolving with phage in a chemostat is used to show that (i) phage predation causes host diversification, and (ii) the diversity that is thus created improves the ability of the host population to optimise growth rates on structured adaptive landscapes. The next section defines the model and methods used. This is followed by results showing the adaptation of the host population on a variety of adaptive landscapes, with and without coevolving phage. The paper concludes with some discussion of the relevance of these findings for artificial and biological evolution.

Model & Methods

Multi-species chemostat model. The model represents the growth and interaction of a diverse community of bacteria and bacteriophage growing in a single-resource chemo-

stat. The model scheme is a variant of a reasonably well-studied type originally formulated for single-species studies of bacteria and bacteriophage growing on a single resource (e.g. (Levin et al., 1977; Bohannan and Lenski, 2000)). Here a multi-species version of the model is used in which mutations can introduce new variants of bacteria and phage, while species are removed (go extinct) when their density falls below a threshold level (Weitz et al., 2005). This creates a simple model in which bacteria and phage phenotypes can evolve by natural selection.

State dynamics for resource concentration R and the density of each host N and phage V population in the chemostat are governed by the following equations:

$$\frac{dR}{dt} = -\omega(R - R_0) - \sum_i \varepsilon \gamma \frac{RN_i \delta_i}{R + K} \quad (1)$$

$$\frac{dN_i}{dt} = -\omega N_i + \gamma \frac{RN_i \delta_i}{R + K} - \sum_j \theta_{ij} N_i V_j \quad (2)$$

$$\frac{dV_j}{dt} = -\omega V_j + \sum_i \beta \theta_{ij} N_i V_j \quad (3)$$

Resource concentration is determined by supply concentration R_0 , flow rate ω , and by the total uptake of resource by all bacterial populations (determined by their growth rate scaled by a resource conversion rate ε). The density N_i of the i^{th} bacterial population is controlled by washout, growth, and mortality from phage. Growth is determined as a function of resource concentration, maximum uptake rate γ , half-saturation constant K , and a strain-specific scaling factor δ_i). The density V_j of the j^{th} phage population is determined by washout and production. Phage production is determined as the sum of production on all available hosts, assuming fixed burst size β and adsorption rate θ_{ij} for phage j on host i . All symbol definitions and parameter values are given in Table 1.

Evolutionary process. Every distinct bacterial genotype h_i and phage genotype v_j is instantiated as a population within the community. Bacterial genotypes are mapped to phenotypic traits for resistance and growth rate. Phage genotypes map to an infection trait. The model assumes that bacteria and phage each evolve within a one-dimensional genetic space, i.e., each distinct genotype can be represented by a point on a line and adaptation occurs by movement along the line. Following mutation, a new population that instantiates the novel phenotype is added to the system. If the density of any population falls below 1 (possible due to the continuous nature of the mathematical abstraction), that population is removed from the system.

The model assumes that mutations are small and occur with low probability for each new cell or virion produced. For bacterial genotype h_i , the instantaneous rate of production of new cells is $\gamma \frac{RN_i \delta_i}{R + K}$ and the probability of mutation

Symbol	Description	Value	Unit
R	Resource concentration	Variable	$\mu g\ ml^{-1}$
N_i	Density of host strain i	Variable	$cells\ ml^{-1}$
V_i	Density of virus strain i	Variable	$virions\ ml^{-1}$
N_{init}	Initial host density	4.6×10^4	$cells\ ml^{-1}$
V_{init}	Initial virus density	8.1×10^5	$virions\ ml^{-1}$
ω	Chemostat dilution rate	0.0033	min^{-1}
R_0	Resource supply concentration	2.2	$\mu g\ ml^{-1}$
ε	Resource conversion rate	2.6×10^{-6}	$\mu g\ cell^{-1}$
γ	Maximum resource uptake rate	0.0123	$\mu g\ min^{-1}$
K	Half-saturation constant	4	$\mu g\ ml^{-1}$
δ_i	Growth scaling for host h_i	Variable	<i>scalar</i>
δ_{min}	Min. growth scaling factor	0.8	<i>scalar</i>
δ_{max}	Max. growth scaling factor	1.2	<i>scalar</i>
ϕ	Maximum adsorption rate	0.104×10^{-8}	$ml(min\ cell)^{-1}$
θ_{ij}	Ads. scaling for v_j on h_i	Variable (range $[0, \phi]$)	<i>scalar</i>
β	Burst size	71	<i>virions</i>
h_i	Genotype of host i	Variable (range $[0, 1]$)	<i>scalar</i>
v_j	Genotype of phage j	Variable (range $[0, 1]$)	<i>scalar</i>
s	Specificity of phage	Manipulated	<i>scalar</i>
M_B	Host mutation rate	0.0001	$cell^{-1}$
M_V	Virus mutation rate	0.0001	$virion^{-1}$
σ_B	Std. dev. of host mut. range	0.005	<i>scalar</i>
σ_V	Std. dev. of virus mut. range	0.005	<i>scalar</i>
Δt	Integration timestep	10	<i>min</i>
T	Simulation duration	10^7	<i>min</i>
L	Chemostat volume	1	<i>ml</i>

Table 1: Model parameters and variable definitions.

of each new cell is M_B , so the number of mutants in each integration timestep for a chemostat of fixed volume L can be calculated. Similarly, the number of mutants of each viral genotype v_j can be calculated using the rate of virion production $\sum_i \beta \phi \theta_{ij} N_i V_j$ and the probability M_V of mutation of each new virion. For each mutation event, the mutant genotype is found by adding a normal deviate to the parental genotype, that is, $h_{mut} = h_i + \mu_h$ (or $v_{mut} = v_i + \mu_v$) where μ_h (μ_v) is a value drawn from a normal distribution with mean 0 and standard deviation σ_B (σ_V).

Infection model. The model uses a similarity-based infection scheme where the likelihood of infection of a host by a phage depends on their genetic ‘similarity’. This scheme captures the basic properties of the matching alleles genetic model and is used to instantiate a density-dependent coevolutionary process. The adsorption coefficient θ_{ij} sets the rate of adsorption to host i by phage j , calculated from the host (h_i) and phage (v_j) genotypes according to:

$$\theta_{ij} = \phi e^{-s(h_i - v_j)^2} \quad (4)$$

where ϕ is the maximum adsorption rate and s is a sensitivity parameter that controls host specificity of phage. This function gives a sigmoidal form with slope determined by s , i.e. tuning the value of s alters the rate of decline in adsorption rate as dissimilarity increases. Every successful adsorption event is assumed to result in infection and instantaneous cell lysis, releasing a burst of β new phage particles.

Host growth rate landscape. To explore the ability of phage-driven diversification to improve host evolutionary search, a bestiary of growth rate functions is used to instantiate different kinds of constraint on the evolutionary

process. In all cases, growth rates perform some mapping of host genotype $h_i \in [0, 1]$ to growth rate scaling factor $\delta_i \in [\delta_{max}, \delta_{min}]$. The landscapes used are:

1. *Flat*. All bacteria have the same growth rate ($\delta_i = 1$).

2. *Single-peak*. A single smooth peak given by:

$$\delta_i = \delta_{min} + (\delta_{max} - \delta_{min})f_1(h_i)$$

where $f_1(h) = e^{-20(h-0.5)^2}$. See Figures 2(a) & 3(a).

3. *Multi-peak*. Multiple smooth peaks given by:

$$\delta_i = \delta_{min} + (\delta_{max} - \delta_{min})f_2(h_i)$$

where:

$$f_2(h) = e^{-100(h-0.2)^2} + 2e^{-50(h-0.5)^2} + 3e^{-50(h-0.8)^2}$$

f_2 is normalised so that the maximum value is 1. See Figures 2(b) & 3(b).

4. *Stepped*. A piecewise linear function of multiple flat plateaus, given by:

$$\delta_i = 0.5 \quad \text{if } 0 \leq h_i < 0.5$$

$$\delta_i = 0.75 \quad \text{if } 0.5 \leq h_i < 0.75$$

$$\delta_i = 1 \quad \text{if } 0.75 \leq h_i \leq 1$$

See Figures 2(c) & 3(c).

5. *Rugged slope*. A linearly increasing function with uniform noise added to introduce ruggedness, given by:

$$\delta_i = \delta_{min} + \frac{1 + 2(h_i + \alpha)}{4}(\delta_{max} - \delta_{min})$$

where α is a uniform random variable drawn from the range $[-d, d]$. See Figures 2(d) & 3(d).

Method. The model is initialised with a single bacterial population and infectious phage, then integrated forward for T minutes using a 4th order Runge-Kutta method with timestep Δt . Data are presented by binning host and phage diversity according to genotype similarity at a resolution of 0.01. Model code, integration and visualisation are performed in MATLAB; code is available on request.

Results

To illustrate the negative density-dependent selection pressure imposed by viral predation, the model was configured so that all host types had the same growth rate (the *flat* landscape), so that the only selectable variation was in host resistance and virus infection traits (Figure 1). The striking feature of this scenario is that there is rapid and sustained diversification of hosts, with correlated diversification of phage. As more host strains enter the system, resource concentrations are drawn down and total bacteria

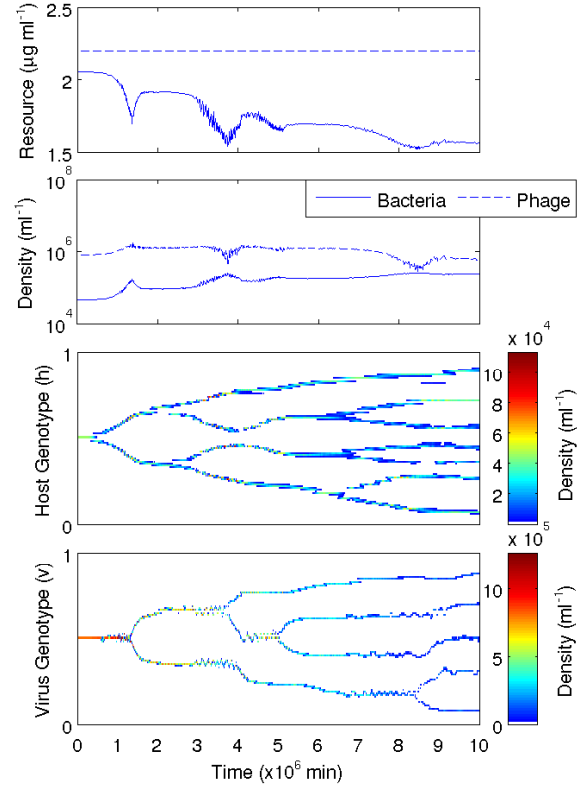


Figure 1: Phage predation causes host diversification. Time-series from a case study using the flat growth landscape ($s = 200$). Plots show: resource concentration (*top*, shown with supply concentration (dashed)), total bacterial and phage density (*middle*), and density of bacteria and phage in genetic space (*bottom*).

density rises. As resource concentrations fall, host growth rates are reduced. Since phage production is proportional to host growth rate (since lysis rate must balance growth rate at steady state), this means that total phage density falls slightly, despite increased host density. The dominant host clusters are relatively evenly distributed across the potential genetic space, reflecting the selective advantage gained from being far enough apart so that each host type is only significantly affected by a single phage strain; moving closer together would expose the host strain to predation by multiple phages and is thus maladaptive. The distance between host clusters, and hence the total genetic variance produced by phage predation, is thus determined by the level of host specificity, i.e. the parameter s setting the decline in adsorption rate with increasing genetic dissimilarity.

The next experiment was to run simulations of (co)evolutionary dynamics on structured growth landscapes,

comparing evolutionary outcomes when hosts evolve without viruses (Figure 2) and with viruses (Figure 3). On the *single-peak* landscape, hosts evolving with and without viruses were able to easily reach the optimum growth rate. However, the diversifying effect of viruses meant that the coevolving hosts were pushed off the peak growth rate (Figure 3(a)), while hosts evolving alone were able to maintain their population close to the peak (Figure 2(a)). The mean growth rate across the evolving host community was close to the maximum achievable value, while mean growth rate for the coevolving hosts was significantly lower.

On the *multi-peak* landscape, hosts evolving alone were able to find the closest local peak to their origin, but became trapped at this local optimum (Figure 2(b)). However, hosts coevolving with viruses were able to reach the global optimum (Figure 3(b)). The coevolving host community was able to escape the local optima, since viral predation caused host diversification into multiple strains that were sufficiently separated in genetic space to cross the fitness valleys between the growth rate optima. Only part of the host community reached the global optimum and several strains remained on the intermediate peak. Therefore the mean growth rate of the host community did not reach the maximum value. Nonetheless the community as a whole had significantly higher growth rates than the host population evolving alone.

On each plateau of the *stepped* landscape, there is no selectable variation in growth rate, so any genetic change with hosts evolving alone is due to drift. However, since populations are large (e.g. $2.74 \times 10^5 \text{ cells ml}^{-1}$ in the example shown), drift does not cause any significant change in genotype frequencies. Thus with hosts alone (Figure 2(c)), no adaptation is observed and the host community remains undiversified at its original position in gene space. When bacteria coevolve with phage (Figure 3(c)), diversification causes the host community to diffuse across the plateau until it reaches the boundary with the next plateau. At that point, selection can allow the community to 'step up' to the next plateau. In this example, although resource competition does not remove all strains on the lower plateau, these strains grow too slowly to support phage.

The *rudded slope* landscape is a linearly increasing slope of growth rate with the addition of uniform noise to create ruddedness. The amplitude d of the noise distribution determines the amount of ruddedness and hence the difficulty of the evolutionary search task; populations must be able to cross small fitness valleys in order to climb the slope towards optimal growth rates. With hosts evolving alone (Figure 2(d)), populations quickly get stuck at a local optimum and are unable to climb the slope. When hosts coevolve with phage (Figure 3(d)), diversification enables the host community to climb the slope effectively so that growth rates are steadily increased.

Discussion

Here a simple model of coevolution between bacteria and bacteriophage was used to explore the impact of coevolution on adaptation of non-phage-related bacterial traits. Coevolution using a similarity-based model of infection (that approximates the operation of the matching-alleles genetic model) showed that phage predation creates negative density-dependent selection that causes diversification of hosts on infection-related traits. When these traits are linked to growth rate, this diversification introduces new variety that can be selected for increased growth rate; that is, it enables evolutionary search. Tests with a variety of growth landscapes show that enhanced evolutionary search promoted by coevolution with phage enables hosts to effectively evolve higher growth rates on structured landscapes.

Coevolutionary diversification of hosts allows more effective search of the genetic space for 'good' growth rate solutions. In effect, it increases *exploration* of the space. However, this comes at a cost of reduced *exploitation* of good solutions once they are discovered. Diversity implies that only one sub-population can occupy the current best location in the search space; all other sub-populations are forced away from the optimum by negative density-dependent selection that prevents convergence. This effect is shown clearly in Figure 3(a). However, on landscapes with multiple local optima and fitness valleys (Figures 3(b) & 3(d)), diversification means that the population as a whole does not get trapped on local optima and can move more effectively towards global optima.

The mechanism identified here might have utility in evolutionary computation, where it might suggest an effective algorithm for search on rugged or multi-peak fitness landscapes. The key component of such an algorithm, here provided by selective phage predation, is negative density-dependent selection; it is this feature of the coevolutionary process that creates diversity and affords enhanced search. This feature could be implemented quite simply in a genetic algorithm, perhaps by imposing a fitness penalty on candidate solutions proportionate to their current representation in the population.

The ability of coevolution with parasites to introduce selectable variation in host traits unrelated to infection depends on the genetic linkage between infection-related traits and other functional traits. There are many cases in nature where parasites attack host traits that have an alternative primary function, e.g. bacteriophage adsorbing to nutrient uptake receptors, or pathogenic fungi inserting hyphae through host plant stomata. Indeed, this should be expected, since any host trait with the sole function of enabling parasitism would be entirely maladaptive and quickly lost by natural selection. Thus it should be expected that mutations with an impact on fitness in the dimension of parasite infection will also impact fitness in the dimension of the trait's original function.

Space constraints in this paper preclude a full analysis of

the sensitivity of these results to model structure and parameters. The model used here is deliberately simple and the number of evolvable parameters has been minimised to reduce the degrees of freedom in the evolutionary process. Alternative formulations might have allowed other traits to evolve, such as half-saturation constant K , burst size β , or maximum adsorption rate ϕ . However, allowing such variation in the current experiment would have obscured the primary result without altering the underlying logic of the argument. The formulation used (evolvable host range of phage, pleiotropy linking evolvable host growth rate and resistance) is sufficient for the current purpose. Two key parameters of the model are mutation range σ_B, σ_V and host specificity of phage s . The results presented are robust to variation in these parameters, so long as mutation range is small and phage have a narrow host range, in relation to the size of structural features in the fitness landscape (e.g. the width of fitness valleys).

The model presented here shows that diversification due to phage predation aids evolutionary search on structured landscapes. On landscapes with multiple peaks, coevolutionary diversification allows hosts to reach global optima; on landscapes incorporating neutral plateaus, coevolutionary diversification causes the population to diffuse and rapidly traverse the plateau; on rugged landscapes, coevolutionary diversification prevents populations becoming trapped by local fitness gradients, so that they can evolve steadily towards optimum growth rates. Biological evolution is far more complex than the simple model presented here, yet biological fitness landscapes are known to often display multiple local optima, neutrality, and ruggedness. Thus it is interesting to hypothesise that diversification due to coevolution with parasites might improve host evolvability in natural systems.

Acknowledgements

I thank four anonymous reviewers. I acknowledge partial financial support from the Leverhulme Trust under the grant “Modelling evolution, ecology and biogeochemistry of marine microbial ecosystems”.

References

- Agrawal, A. and Lively, C. (2002). Infection genetics: gene-for-gene versus matching-alleles and all points in between. *Evolutionary Ecology Research*, 4:79–90.
- Agrawal, A. and Lively, C. (2003). Modeling infection as a two-step process combining gene-for-gene and matching alleles. *Proc. Roy. Soc. Lond. B*, 270:323–334.
- Avrani, S., Wurtzel, O., Sharon, I., Sorek, R., and Lindell, D. (2011). Genomic island variability facilitates *prochlorococcus*-virus coexistence. *Nature*, 474:604–608.
- Bohannan, B. and Lenski, R. (2000). Linking genetic change to community evolution: insights from studies of bacteria and bacteriophage. *Ecology Letters*, 3:362–377.
- Brockhurst, M., Morgan, A., Fenton, A., and Buckling, A. (2007). Experimental coevolution with bacteria and phage: The *pseudomonas fluorescens*— ϕ 2 model system. *Infection, Genetics and Evolution*, 7:547–552.
- Buckling, A. and Rainey, P. (2002). Antagonistic coevolution between a bacterium and a bacteriophage. *Proceedings of the Royal Society: Biological Sciences*, 269(1494):931–936.
- Fenton, A., Antonovics, J., and Brockhurst, M. (2012). Two-step infection processes can lead to coevolution between functionally independent infection and resistance pathways. *Evolution*. Early edition.
- Fenton, A., Antonovics, J., and Brockhurst, M. A. (2009). Inverse gene-for-gene infection genetics and coevolutionary dynamics. *American Naturalist*, 174:E230–E242.
- Hall, A., Scanlan, P., Morgan, A., and Buckling, A. (2011). Host-parasite coevolutionary arms races give way to fluctuating selection. *Ecology Letters*, 14(7):635–642.
- Lennon, J., Khatana, S., Marston, M., and Martiny, J. (2007). Is there a cost of virus resistance in marine cyanobacteria? *ISME J.*, 1(4):300–312.
- Levin, B., Stewart, F., and Chao, L. (1977). Resource-limited growth, competition, and predation: A model and experimental studies with bacteria and bacteriophage. *The American Naturalist*, 111:3–24.
- Marston, M., Pierciey, F., Shepard, A., Gearin, G., Qi, J., Yandava, C., Schuster, S., Henn, M., and Martiny, J. (2012). Rapid diversification of coevolving marine *synechococcus* and a virus. *PNAS*. Early edition.
- Odling-Smee, F. J., Laland, K. N., and Feldman, M. W. (2003). *Niche Construction: The Neglected Process in Evolution*. Princeton University Press.
- Rodriguez-Valera, F., Martin-Cuadrado, A.-B., Rodriguez-Brito, B., Pasic, L., Thingstad, T. F., Rohwer, F., and Mira, A. (2009). Explaining microbial population genomics through phage predation. *Nat. Rev. Microbiology*, 7(11):828–836.
- Thingstad, T. (2000). Elements of a theory for the mechanisms controlling abundance, diversity, and biogeochemical role of lytic bacterial viruses in aquatic systems. *Limnology and Oceanography*, 45:1320–1328.
- Thingstad, T. and Lignell, R. (1997). Theoretical models for the control of bacterial growth rate, abundance, diversity and carbon demand. *Aquatic Microbial Ecology*, 13:19–27.
- Thompson, J. (2005). *The Geographic Mosaic of Coevolution*. University of Chicago Press, Chicago, USA.
- Weitz, J., Hartman, H., and Levin, S. (2005). Coevolutionary arms races between bacteria and bacteriophage. *PNAS*, 102:9535–9540.
- Williams, H. T. P. and Lenton, T. M. (2008). Environmental regulation in a network of simulated microbial ecosystems. *PNAS*, 105(30):10432–10437.
- Woolhouse, M., Webster, J., Domingo, E., Charlesworth, B., and Levin, B. (2002). Biological and biomedical implications of the coevolution of pathogens and their hosts. *Nature Genetics*, 32(4):569–577.

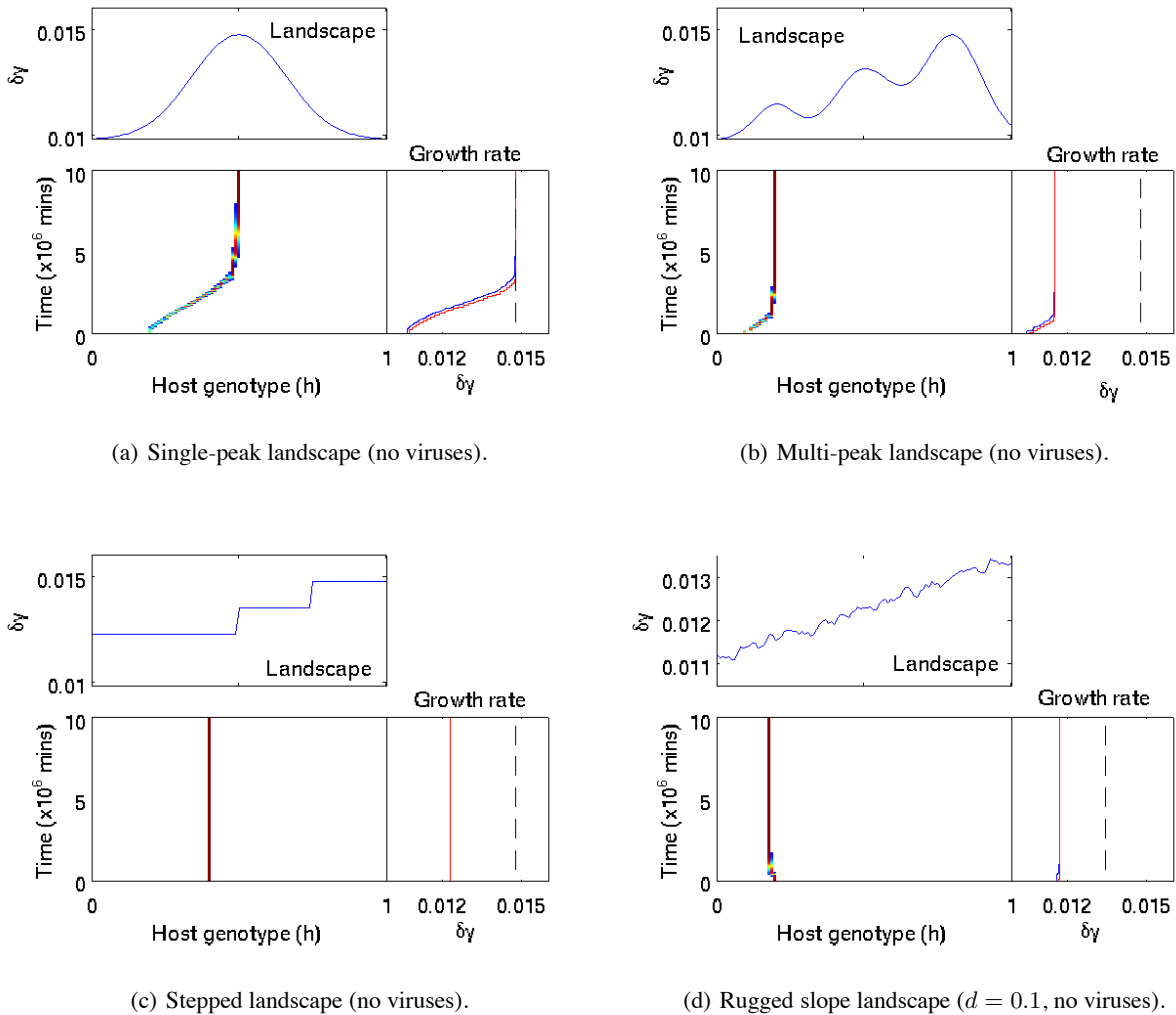
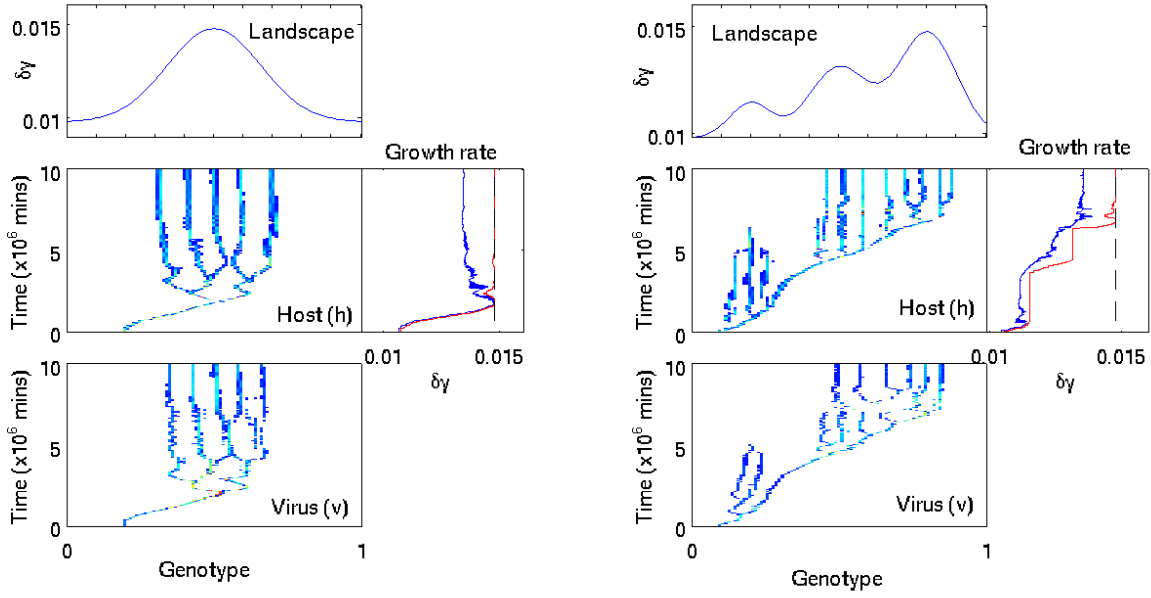
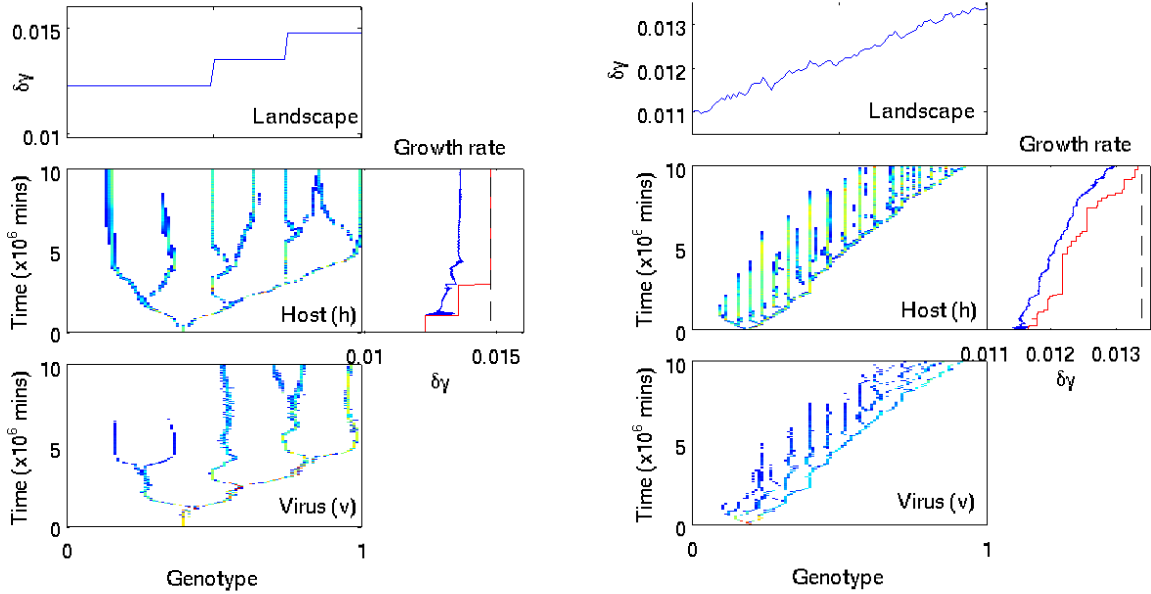


Figure 2: Case study simulations showing host evolutionary dynamics on various growth landscapes in the absence of viruses. Plots show the growth landscape (*top*, given as the value of $\delta_i\gamma$ for all possible host genotypes h_i), the distribution over time of hosts in genetic space (*middle*), and the observed growth rates over time of the host community (*right*, shows mean (blue), actual maximum (red), potential maximum (dashed)).


 (a) Single-peak landscape ($s = 200$).

 (b) Multi-peak landscape ($s = 500$).

 (c) Stepped landscape ($s = 100$).

 (d) Rugged slope landscape ($d = 0.1, s = 1000$).

Figure 3: Case study simulations showing coevolutionary dynamics and host growth rate adaptation on various growth landscapes. Plots show the growth landscape (*top*, given as the value of $\delta_i\gamma$ for all possible host genotypes h_i), the distribution over time of hosts and viruses in genetic space (*middle and bottom*), and the observed growth rates over time of the host community (*right*, shows mean (blue), actual maximum (red), potential maximum (dashed)).

Sexual Selection, Resource Distribution, and Population Size in Synthetic Sympatric Speciation

Mark Woehrer¹, Dean Hougen¹, and Ingo Schlupp²

¹School of Computer Science

²Department of Biology

University of Oklahoma, Norman, OK 73072

markwoehrer@ou.edu

Abstract

Speciation is one of the most fundamental and important processes in evolutionary biology, resulting in the panoply of biological diversity found in the natural world. Speciation likewise has profound implications for artificial life, evolutionary computation, and evolutionary robotics, yet a great many aspects of it remain unexplored. Traditionally, speciation was mainly viewed as taking place allopatrically. More recently, sympatric speciation, which does not require geographic isolation, has been studied. Sympatric speciation raises a number of interesting questions with regard to how and why sympatric populations diverge, some of which we address with a 2x2x2 factorial study that considers the factors of sexual selection, resource distribution, and population size. Our hypotheses were evaluated using a synthetic environment inspired by life on the Galápagos Islands. In particular, the wet and dry season dynamics were modeled to produce the intense selection pressure found there. Our results provide direct evidence for the importance of both female mate choice and resource availability on speciation. They also suggest that the greater stability afforded by larger populations can lead to subpopulations between which gene flow is reduced.

Introduction

We are interested in understanding both “life-as-we-know-it” and “life-as-it-might-be.” The natural world possesses a rich biodiversity brought about through biological evolution, a process we are keenly interested in understanding. Likewise, we are greatly interested in understanding and interpreting the possible mechanisms for the evolution of diversity in synthetic life. In particular, we would like to create environments in which synthetic ecological webs promote the divergence of existing forms into multiple new ones.

In most artificial life and evolutionary computation studies there is a single population, within which all members freely interbreed (typically using a recombination operator known as crossover) or none of which interbreed (typically variation is introduced through different types of mutation) (Bedau, 2003; De Jong, 2006). In nature, by contrast, there are countless population-like units within which there is significant interbreeding yet between which there is little or no breeding. These units are often known as *species* and the

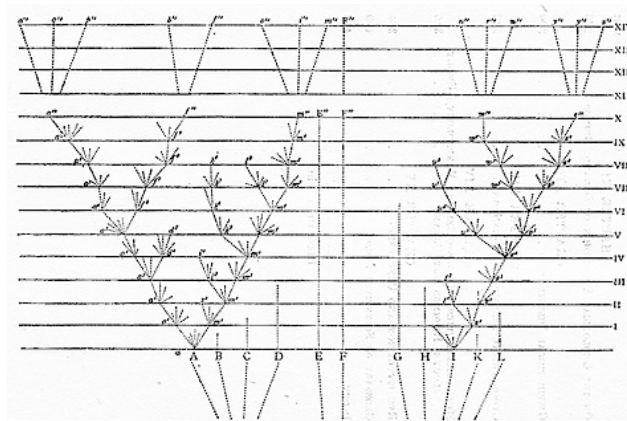


Figure 1: Evolution in action—Darwin’s conceptual diagram of speciation (Darwin, 1859).

division of a single interbreeding population into multiple distinct such populations is known as *speciation*. Species may provide a wealth of diversity in an environment by filling distinct niches. Darwin’s concept of speciation is shown in the only figure he included in his seminal work *On the Origin of Species* (1859) (reproduced here as Figure 1).

Artificial mechanisms could be used to subdivide artificial populations more or less completely, and many such mechanisms have been proposed including crowding (De Jong, 1975), niching (Goldberg, 1989; Horn et al., 1994), tagging (Spears, 1994) imposing a population topology (Sarma, 1998) and using islands (Whitley et al., 1999). All of these approaches have merits for their intended applications but are not entirely appropriate for ours. In particular, none of these allow for new niches to arise based on the behavior (e.g., resource use) of groups within the populations. Other niching approaches (e.g., Tomko et al. 2011) are aimed at evolving a collection of cooperating partial solutions to a problem, rather than evolving independent populations.

Rather than impose upon the algorithm population divisions, or other mechanisms to promote population divisions (Gras et al., 2009; Aspinall and Gras, 2010), we prefer to allow speciation to occur based on interactions between in-

dividuals within an environment where the interactions arise from mechanisms and actions inherently necessary for the individuals' survival and procreation.

For evolutionary biologists, speciation is one of the most fundamental processes. It is the way biodiversity is generated and a phenomenon that has intrigued biologists since Darwin's time (Darwin, 1859). Traditionally, speciation was mainly viewed as allopatric or geographic speciation. Here species are separated into at least two distinct and geographically isolated units, evolve independently into separate species, then cannot interbreed even if they come into contact again. More recently, another mechanism of speciation has been studied that does not depend on geographic isolation. *Sympatric speciation* occurs when budding species, living together in the same area, split into two or more populations exploiting different niches. For example, in the Galápagos Islands, small populations of finches with different beak sizes (different species) are known to inhabit the islands (Grant and Grant, 1987). With allopatric speciation alone it would be likely that each island would contain a different species of its own but, in fact, different islands contain multiple species living and breeding on them.

The existence and possibility of sympatric speciation processes have long been unclear, but recent theoretical, observational, and experimental studies have made it clear that sympatric speciation is more common than previously thought (Coyne and Orr, 2004). Various mechanisms, such as local abiotic conditions (Tobler et al., 2008; Riesch et al., 2010), can lead to population divergence within a habitat. On the Galápagos, the harsh dry season, in which food abundance drops and the birds forage on increasingly scarce seeds of different sizes, appears to provide one mechanism.

Biologists have identified behavior as an important mechanism causing divergence, in particular female mate preference (Seehausen et al., 1997; Seehausen and van Alphen, 1999; Kraaijeveld et al., 2011). Over the last decade there has been considerable work that provides support for behavior being an important factor in sympatric speciation (Coyne and Orr, 2004). Assortative mating, in which females prefer to mate with males similar to themselves, has been identified as a key element in the development of sympatric speciation (Seehausen and van Alphen, 1999; Kraaijeveld et al., 2011).

The combination of divergent selection (e.g., natural selection during the dry season) and assortative mating acting on the same trait (e.g., beak size) results in a *magic trait*, "a trait subject to divergent selection and a trait contributing to non-random mating that are pleiotropic expressions of the same gene(s)" (Servedio et al., 2011).

By using sympatric speciation, ALife researchers can support multiple species without placing physical barriers in the environment (Yaeger, 1994). In recent work, the need for simpler environments supporting sympatric speciation has been identified (Murdock and Yaeger, 2011). Using assortative mating in agent-based simulations allows for the natu-

ral emergence of diversification and hence speciation. "This common feature suggests that the evolution of biodiversity may be driven not simply by natural-selective adaptation to ecological niches, but by subtle interactions between natural selection and sexual selection" (Todd and Miller, 1997).

Speciation is a very time-consuming process. What is currently poorly understood in evolutionary biology is how often incipient divergence will actually lead to a speciation event and how often the process is aborted. This question has recently been receiving increased attention both theoretically (Bolnick, 2011) and empirically (Vonlanthen et al., 2012). Since the actual process is slow, the true dynamics are difficult to observe or study experimentally, but can be studied in simulations. One of the specific aims of this study was to use an ecological simulator to investigate divergence and speciation under a number of conditions. Our model is based on Galápagos finches but the results are broadly applicable where natural selection and/or mate choice appear.

Hypotheses

The first question we address is whether the existence of differences in the distribution of resources will lead to divergence and speciation. In our simulations we address this by testing two different resource distributions (simulated as seed distributions), bimodal and uniformly random. The second question we address is whether female mate preference will strengthen divergence. Here, we provide two different forms of mate selection, assortative and random. For these different experimental conditions we formed four independent hypotheses:

H₁ For bimodal seeds and assortative mating (BSAM) we expect to find speciation. We reasoned that the bimodal seed distribution provides the environmental structure needed to support two species along with assortative mating which ensures that reproduction produces viable offspring.

H₂ For bimodal seeds and random mating (BSRM) we expect to find directional selection but no speciation. A moderate beak size is rather untenable because there are few moderately sized seeds. We therefore predicted that the population would converge to either small or large beaks, matching either mode of the seed size distribution.

H₃ For uniform seeds and assortative mating (USAM) we may find speciation. There was no clear prediction on what the outcome would be but speciation seemed possible, because assortative sexual selection could drive the speciation process despite the fact that there were no clear environmental niches to occupy.

H₄ For uniform seeds and random mating (USRM) we do not expect to find speciation. Since there were no resource niches around which species could form and no sexual selection to drive speciation, we predicted no speciation.

The third question we address is whether population size will affect speciation. Since we do not directly control population size for the finches, this question was addressed by

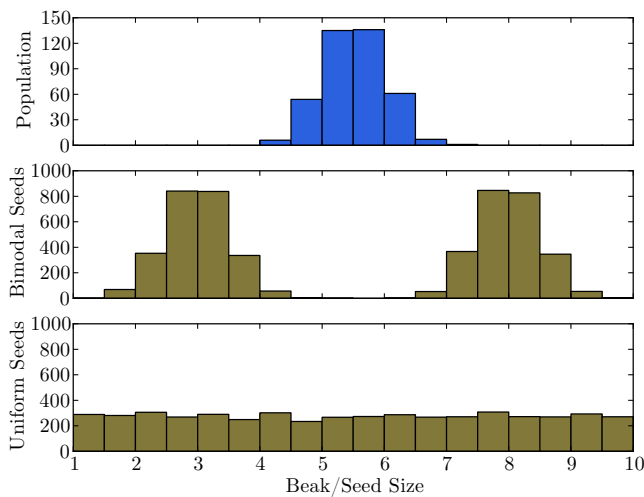


Figure 2: An illustration of the initial population beak size (top) and seed size distributions for the 1x seeds case (middle and bottom for bimodal and uniform random seeds, respectively).

varying the number of seeds provided at the start of the dry season. We had two seed conditions, low and moderate, where the moderate condition had ten times as many seeds as the low condition. This led to a fifth hypothesis:

H₅ For a moderate increase in the number of seeds we expect to see a corresponding increase in population size and a greater stability in all four cases. We predicted that, even with the increased stability, we would not see differences with respect to the presence or absence of speciation in any of the four cases.

These three questions, with two conditions tested for each question, resulted in our 2x2x2 factorial study.

Methods

To test these hypotheses, we developed an artificial island—a square region 100 x 100 units, containing two types of simulated objects: birds and seeds. The birds have the following individual properties: age, beak size, energy level, and gender. The seeds have a specific energy, location, and size. The birds have two additional constraints—a maximum energy capacity of two units and a lifespan of four years. Using this island, we conducted numerous repetitions for each of the four experimental conditions for up to 1000 generations. A repetition can end before 1000 generations if complete extinction occurs. At each generation of a run (a particular repetition) we logged data related to the individuals and the seeds; in particular, we recorded the following data for each individual: age, beak size, energy, gender, and mating count. In addition we recorded a unique identifier for new offspring along with the identifiers of the parents. The data recorded for each seed includes energy and location.

Two different seed distributions are used to model the

available food resources as shown in Figure 2. The bimodal seed distribution (means 3 and 8, variance 0.5) represents an environment that contains two distinct seed sizes with a limited amount of variation. Conversely, the uniform seed distribution (1 to 10) models an environment in which there is no distinction with regard to abundance for any given seed size. The initial population size for each experimental run was 400 individuals possessing moderately sized beaks (mean 5.5, variance 0.5) as shown in Figure 2. The initial population had a 1:1 sex ratio.

The extended dry season on the Galápagos Islands is modeled as an interval lasting 100 days. On each day the individuals search the island looking for seeds. As each day passes there are fewer and fewer seeds on the island—the seeds are present at the beginning of the dry season and are gradually consumed by the individuals. For the small population condition, we started the dry season with 5,000 seeds. For the moderate population condition, we started with ten times as many. These conditions are therefore called the 1x and 10x seed conditions, respectively. We conducted 48 repetitions for each of the four seed/mating combinations using the 1x seed condition and 24 repetitions for the 10x seed condition.

During each day of natural selection the individuals feed in random order—only one feeding attempt per day. To simulate the feeding process, an individual first picks a random region, 10x10 units in size, to search. In this region the individual will look for seeds that are compatible with its beak size. An individual can consume seeds plus or minus one unit from its beak size. For example, an individual with a beak size of 4.2 can only select seeds within the range of 3.2 to 5.2. From these acceptable seeds, an individual selects one seed at random and consumes all of its energy. The exact amount of energy contained in each seed varies randomly from zero to two units (uniformly random). The cost for search is 0.1 units of energy—considerably less than the energy gained from an average seed. Note that the energy level of an individual is decreased even if no seed is consumed. After accounting for the cost for searching, the energy level of the individual is examined and if it falls below zero that individual is removed from the population. At the end of 100 days, the dry season ends and a season of abundance begins, during which all individuals who survived the dry season's harsh natural selection process may attempt to mate and produce offspring. The first step in this process is sexual selection.

During sexual selection all females are allowed to select a male and produce offspring. In this simulation the female can show two possible mating behaviors—assortative or random mating—as determined by the experimental conditions of the run. For assortative mating, the female is choosy with respect to the mate she selects—she will only choose a male that is plus or minus one unit from her own beak size. If there is more than one acceptable mate, the female chooses one of those males at random. To limit the influence of a sin-

gle male, and to account for the limited energy males have for courting females, males are only allowed to mate five times per breeding season. For random mating, a female selects one of the males in the population at random. Note that random mating is just a special case of assortative mating where the acceptable beak size is large enough to encompass all males for any given female.

During the reproduction phase, the female mates with the selected male and produces an offspring. This offspring has a beak size that is the average of its parents' plus a small amount of random mutation in the form of Gaussian noise (mean 0, variance 0.2). The gender of the new offspring is determined randomly and the energy level is set to zero.

After reproduction the age of each individual in the population is incremented by one and individuals older than four are removed. The remaining seeds are removed and a new supply of seeds is added—a new dry season begins.

Results

The data is presented in a series of what we have termed *phenogenealogic trees*, which illustrate evolution in action. The trees show each individual in the population plotted with respect to beak size and generation. We connect each individual to its parents using lines forming a tree. A marker is drawn for all the individuals in a given generation. Females are represented as circles, males as squares. Two different colors are used to represent individuals who survived natural selection (blue/dark) and individuals that perished (green/light). Individuals who perish during natural selection are superimposed on individuals who survived.

1x Seeds

The phenogenealogic trees for BSAM, USAM, BSRM, and USRM Run 1 are shown in Figure 3a–3d as prototypical examples. The average initial population sizes (generation 1, after natural selection) for the bimodal random seed and uniform random seed cases are 11.1 and 5.82 respectively. The average final population sizes (generation 1000, after natural selection) for BSAM, USAM, BSRM, and USRM are 43.5, 41.31, 26.0, and 18.4 respectively. When extinction occurred, it was more often due to a lack of females than a lack of males, since a single male can mate with up to five females in a single breeding season, resulting in five new offspring, whereas a single female can only produce a single offspring regardless of her mating activities.

BSAM: It is clear that once divergence occurs no interbreeding takes place between the two branches. The populations for the left and right branches have an average beak size of three and eight and remain stable up to 1000 generations. Stability here refers to the fact that the populations do not go extinct. Such speciation was observed in 31 out of 48 repetitions. In the other 17 repetitions, one of the two populations went extinct primarily due to the lack of females. In

13 of these cases, branch extinction took place before generation 10. Complete extinction did not occur in any of the repetitions.

USAM: The defining characteristic for USAM with small populations is the repeated branching and merging—with more branching than merging. Here, populations are not fixed entities; when a population goes extinct another population moves in to fill the niche. Boundary effects appear to be present in that populations do not occupy the lower and upper size limits of the food supply. The same pattern is repeated throughout each repetition—significant die-off in the center of a given population followed by divergence and/or possible extinction. For example, just after generation 400 the branch with an average beak size near 3.0 splits into two populations which then merge back together a few generations later. Stability of new branches is not guaranteed. For example, the branch with an average beak size near 2.25 goes extinct just after generation 200. This is most likely due to a sex ratio imbalance, which is a result of small population sizes. Also, something akin to a genetic drift component is present which causes a random wobble in each subpopulation. We identified the number of populations in the final generation of each repetition. In a single repetition we found one population, in five repetitions we found two populations, in 19 repetitions we found three populations, and in 10 repetitions we found four populations. The remaining 13 repetitions ended in complete extinction.

BSRM: The results are very similar to the BSAM case except that only one population is supported. A single population with an average beak size of three is clearly stable up to generation 1000. As is the case for BSAM the high amount of variability in the individuals is clearly visible. The convergence to a single population with an average beak size of three or eight is a defining feature for BSRM. In 41 of the repetitions a single population, centered on one of the two distinct seed sizes, is present in the final generation. In the remaining seven repetitions, there is complete extinction by generation 1000. In three of these cases, complete extinction took place before generation 10.

USRM: The defining characteristic for USRM is a single population, stable up to generation 1000 with a central green/light band indicating a large die-off in the center of the population. Also, there is no branching as seen in the USAM case. Wobble in the average population beak size, a result of a process akin to genetic drift, is clearly visible. In 13 repetitions the population went extinct before generation 1000. In six of the cases, complete extinction takes place before generation 10, due to the lack of males (three cases) or females (three cases) in the population. In the seven remaining repetitions (after generation 10) the population went extinct entirely due to the lack of females.

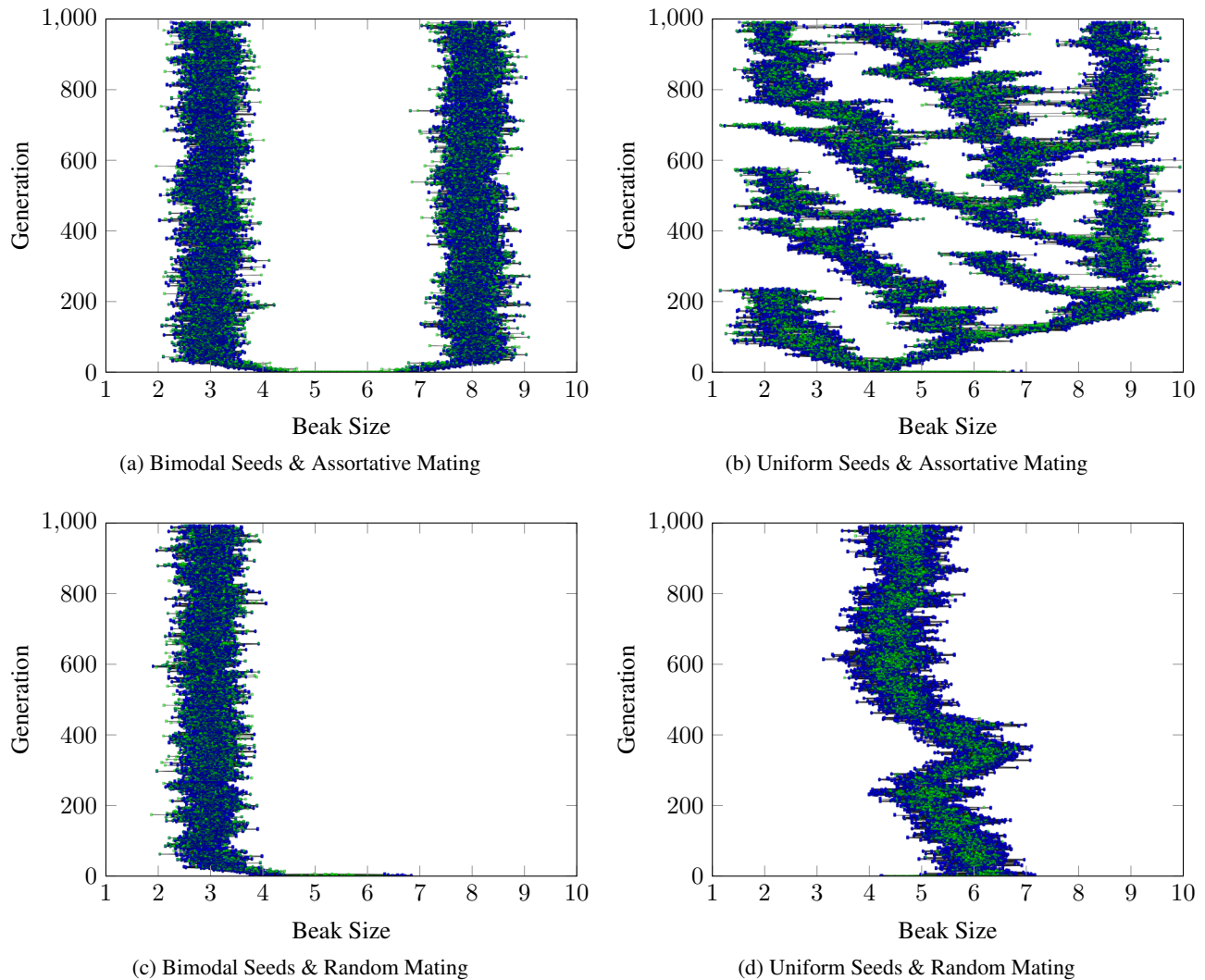


Figure 3: An illustration of the four combinations for the small population condition (5,000 seeds). (a) For the bimodal seeds and assortative mating combination, two populations are clearly present. (b) In the uniform seeds and assortative mating case, there is considerable branching and some merging. (c) When the mating behavior is changed from assortative to random mating, only one of the niches is exploited. (d) The chaotic branching found in (b) is reduced to a single population. Here there is significant die-off near the center of the population during each generation but divergence does not occur because mating is random within the whole population.

10x Seeds

The phenogenealogic trees for the 10x seed case are shown in Figure 4. The overall results are the same as the 1x seed case except for the USAM combination. In all combinations the population sizes were larger due to increased food resource and therefore more stable (there were no complete extinctions before generation 1000). The average initial population size (generation 1, after natural selection) for the bimodal and uniform seed cases are 49.9 and 139 respectively. The average final population size (generation 1000, after natural selection) for BSAM, USAM,

BSRM, and USRM are 541, 546, 280, and 208 respectively. For USAM (Figure 4b), the chaotic branching and merging found in the corresponding 1x case is absent. Instead, four populations are formed around generation 100 and remain stable up to generation 1000. A significant amount of interbreeding takes place between the adjacent branches.

Discussion

In this study we demonstrated the usefulness of our framework and its overall utility when applied to the finch studies of the Galápagos Islands. By focusing on one phenotypic

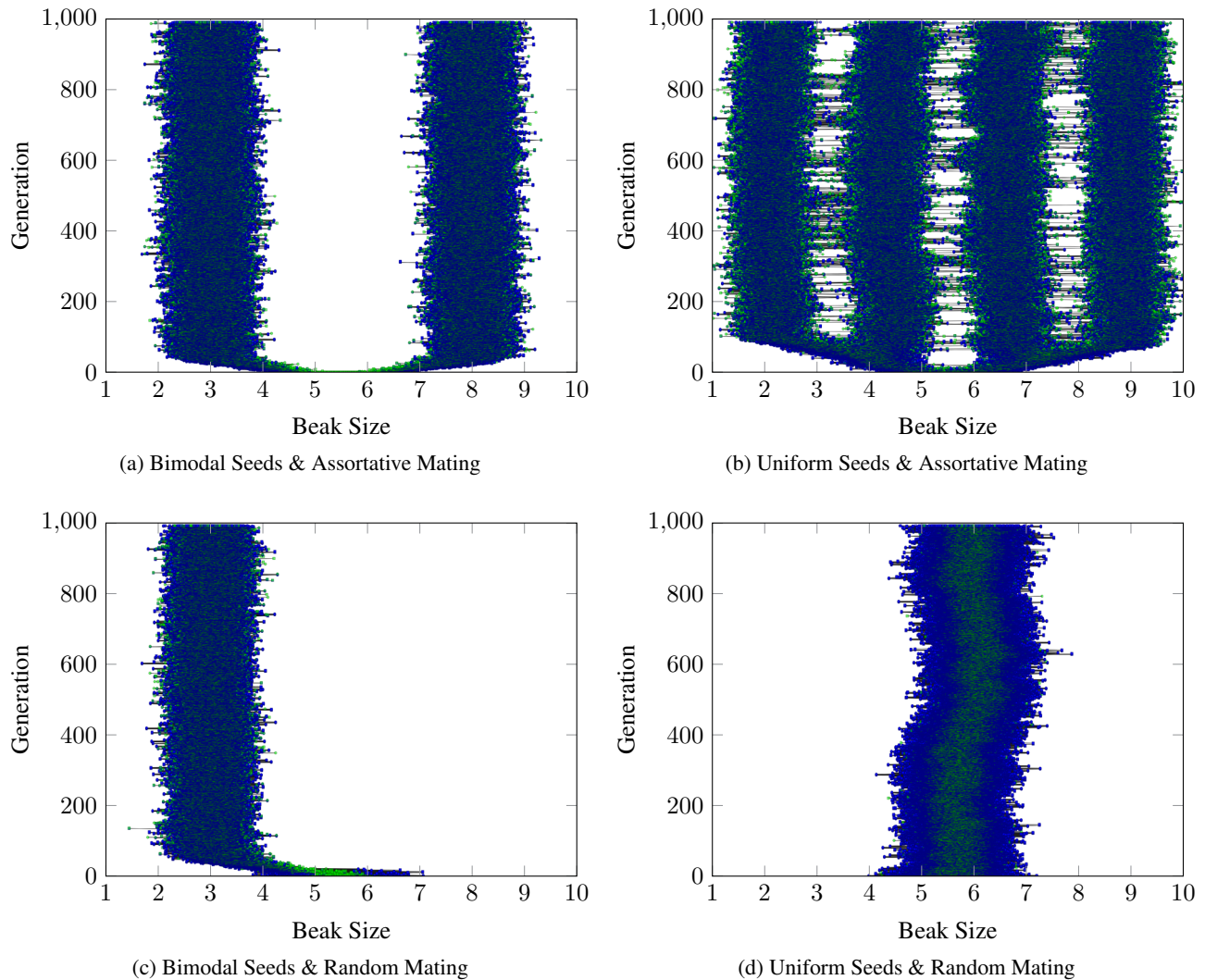


Figure 4: An illustration of the four combinations for the moderate population condition (50,000 seeds). In all four cases the increased population size is clearly visible. (a) For the bimodal seeds and assortative mating combination there is qualitatively no change from the 1x case. (b) The results for uniform seeds and assortative mating are substantially different from the 1x case. The intricate branching patterns have been replaced by four stable populations. (c) The bimodal seeds and random mating combination is qualitatively similar to the corresponding 1x case. (d) Likewise, the uniform seeds and random mating results are similar to the 1x counterpart except for a decreased drift-like component.

trait we have shown how a highly variable trait along with the process of natural selection and sexual selection can lead to speciation and therefore diversity.

We addressed interesting questions related to mate choice in our ecological simulation. The first question regarding the ability of the population to track resources was addressed. We found that our simulated bird populations evolved specialized beaks for the food resources available. The second question regarding the role of female mate preferences was addressed. We found that sexual selection based on assortative mating was necessary for speciation in our simulations.

Our results showed no divergence for the USRM combination, generally maintaining one lineage that did not shift much in expression of the trait over time. Similarly, BSRM generally led to a single species but with a shift of the mean beak size. These results may be surprising from a biological perspective because they leave substantial resources unused. However, they reflect the restrictions put on the experiments. For example, because the simulated finches are not allowed to evolve assortative mating in the random mating conditions, that mechanism for speciation is removed and ecological niches are left unfilled.

More interestingly, we found very rapid divergence in BSAM. Very quickly two populations evolved tracking the two available seed sizes. This rapid divergence was accentuated by assortative mating. The most interesting case is when assortative mating is combined with uniform random seeds (USAM). We hypothesized that we might see something akin to speciation supporting a given number of populations. This was supported by the data we collected. However, we did not anticipate the diverse branching and less frequent merging seen across the beak/seed range for the 1x seeds case. We observed long term stability in these new populations, remaining distinct for fifty generations or more. Nonetheless, the observed patterns are characterized by significant interplay between lineages, multiple lineages going extinct, and overall the most complex trees. The case was quite different for USAM in the 10x seed case. In this case, we observed rapid divergence but, rather than the complex splitting and merging patterns seen in the 1x seed case, the resulting four populations in the 10x case stayed quite stable, maintaining consistent means and avoiding extinction, but were not entirely distinct from one another with frequent hybridization observed.

In our study, the interplay of natural selection and sexual selection leads to speciation. Natural selection causes the initial die-off in the center of the beak size distribution, increasing the variance and essentially forming two new populations, and selection keeps them apart. Under conditions with more resources (10x seeds) we observed wider populations overall, but the qualitative patterns we found were very similar to the 1x seeds case. One important difference though was that in USAM four apparent lineages were supported, which appear distinct in the trees (Fig. 4b), but are connected by massive hybridization, suppressing true speciation. It appears that the more relaxed ecological conditions used here do not favor complete divergence whereas harsher conditions and smaller population sizes do.

Our results are important to biologists because they show how deceptive viewing speciation phenomena over a limited time can be: if one had a study lasting 200 generations in our USAM example, one would conclude that speciation has occurred and produced three distinct lineages. Without major changes in ecology, however, this situation changes drastically and eventually leads to four distinct lineages after 1000 generations. One interpretation of this finding is that early stages of divergence are more labile than currently thought and can collapse again for many generations. Our finding is in agreement with other recent work on sticklebacks (Bolnick, 2011) and whitefish (Vonlanthen et al., 2012).

Overall, our findings are congruent with earlier empirical studies that indicated an important role of female choice (Seehausen and van Alphen, 1999; Boake, 2000; Boughman, 2001; Bleay and Sinervo, 2007; The Marie Curie SPECIATION Network, 2011). It is also noteworthy that sexual selection using beak size has been implicated in other finches,

too (Slabbekoorn and Smith, 2000).

Interestingly, small population size is of great importance in our simulations and generally favors divergence (but also leads to random extinctions). The finch populations on which we are basing our simulations have small population sizes so there is a biological basis for this discussion. Therefore, our simulation results may have an impact on biologists studying small populations sizes in general (Grant and Grant, 2011). The possibility of extinction is always a concern with small population sizes and is of particular importance in the study of speciation.

More investigation is needed but larger population sizes may affect the outcome for BSRM. If larger population sizes are tested—at least two orders of magnitude larger than our smallest case—we might see two populations supported. We saw some hints of support for this in our 10x case.

Our study provides key insights to the ALife community. By coupling natural selection and sexual selection by using so-called magic traits (Servedio et al., 2011), we demonstrated how population diversity can be generated and maintained. In this work, we did not focus too much on how to identify species. Instead, we produced a synthetic environment in which clustering and hence speciation was an emergent property of our system. Although we validated our framework using the finch studies of the Galápagos Islands, the results could be applied broadly where natural selection and/or sexual selection operate. We find the interplay between these two selection methods to be a tantalizing prospect for the evolution of meaningful diversity.

Future Work

We believe that this research can be extended in a number of important ways. We made a significant contribution in the qualitative analysis of the data with our phenogenealogic tree. We would like to extend this work to a quantitative analysis of the phenomena observed. One possibility is the automatic identification of different species (Murdock and Yaeger, 2011). Manually identifying the different species in a given population is a tedious process and especially difficult when the subpopulations are not well defined. With this new capability we could gather new statistics for each population, such as size and gender ratio, which may be useful in predicting extinctions.

Although we have used key principles from biology in the design of our simulation, we are interested in validating our results with actual finch data. Given that our framework employs an agent-based model, it is well-suited to incorporating empirically measurable parameters.

Acknowledgments

This research was funded by the Potentially Transformative Research, Scholarship and Creative Activity Program of the Office of the Vice President for Research and The Research Council, University of Oklahoma.

References

- Aspinall, A. and Gras, R. (2010). K-means clustering as a speciation mechanism within an individual-based evolving predator-prey ecosystem simulation. In *Proceedings of the 6th International Conference on Active Media technology, AMT'10*, pages 318–329, Berlin, Heidelberg. Springer-Verlag.
- Bedau, M. A. (2003). Artificial Life: organization, adaptation and complexity from the bottom up. *Trends in Cognitive Sciences*, 7(11):505–512.
- Bleay, C. and Sinervo, B. (2007). Discrete genetic variation in mate choice and a condition-dependent preference function in the side-blotched lizard: implications for the formation and maintenance of coadapted gene complexes. *Behavioral Ecology*, 18(2):304.
- Boake, C. R. B. (2000). Flying apart: mating behavior and speciation. *BioScience*, 50(6):501–508.
- Bolnick, D. I. (2011). Sympatric Speciation in Threespine Stickleback: Why Not? *International Journal of Ecology*, 2011:1–15.
- Boughman, J. W. (2001). Divergent sexual selection enhances reproductive isolation in sticklebacks. *Nature*, 411(6840):944–948.
- Coyne, J. A. and Orr, H. A. (2004). *Speciation*. Sinauer Associates Sunderland, MA.
- Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. Mentor Reprint, 1958, NY.
- De Jong, K. A. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, Ann Arbor, MI, USA.
- De Jong, K. A. (2006). *Evolutionary Computation: A Unified Approach*. MIT Press.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional.
- Grant, P. R. and Grant, B. R. (1987). Sympatric speciation and Darwin's finches. *Speciation and its consequences*. Sinauer, Sunderland, MA, pages 433–457.
- Grant, P. R. and Grant, B. R. (2011). *How and why species multiply: the radiation of Darwin's finches*. Princeton University Press.
- Gras, R., Devaurs, D., Wozniak, A., and Aspinall, A. (2009). An individual-based evolving predator-prey ecosystem simulation using a fuzzy cognitive map as the behavior model. *Artificial Life*, 15(4):423–463.
- Horn, J., Nafpliotis, N., and Goldberg, D. E. (1994). A niched Pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, pages 82–87. IEEE.
- Kraaijeveld, K., Kraaijeveld-Smit, F. J. L., and Maan, M. E. (2011). Sexual selection and speciation: the comparative evidence revisited. *Biological Reviews of the Cambridge Philosophical Society*, 86(2):367–377.
- Murdock, J. and Yaeger, L. (2011). Identifying species by genetic clustering. In *Proceedings of the European Conference on Artificial Life*, pages 565–572, Paris, France. MIT Press.
- Riesch, R., Oranth, A., Dzienko, J., Karau, N., Schießl, A., Stadler, S., Wigh, A., Zimmer, C., Arias-Rodriguez, L., Schlupp, I., and Plath, M. (2010). Extreme habitats are not refuges: poeciliids suffer from increased aerial predation risk in sulphidic southern Mexican habitats. *Biological Journal of the Linnean Society*, 101(2):417–426.
- Sarma, J. A. (1998). *An Analysis of Decentralized and Spatially Distributed Genetic Algorithms*. PhD thesis, George Mason University, Fairfax, VA, USA.
- Seehausen, O., Alphen, J. J. M., and Witte, F. (1997). Cichlid fish diversity threatened by eutrophication that curbs sexual selection. *Science*, 277(5333):1808.
- Seehausen, O. and van Alphen, J. J. M. (1999). Can sympatric speciation by disruptive sexual selection explain rapid evolution of cichlid diversity in Lake Victoria? *Ecology Letters*, 2(4):262–271.
- Servedio, M. R., Van Doorn, G. S., Kopp, M., Frame, A. M., and Nosil, P. (2011). Magic traits in speciation: 'magic' but not rare? *Trends in Ecology & Evolution*, 26(8):389–397.
- Slabbekoorn, H. and Smith, T. B. (2000). Does bill size polymorphism affect courtship song characteristics in the African finch *Pyrenestes ostrinus*? *Biological Journal of the Linnean Society*, 71(4):737–753.
- Spears, W. M. (1994). Simple subpopulation schemes. In *Proceedings of the Evolutionary Programming Conference*, volume 3, pages 296–307.
- The Marie Curie SPECIATION Network (2011). What do we need to know about speciation? *Trends in Ecology & Evolution*, 27(1):27–39.
- Tobler, M., DeWitt, T., Schlupp, I., García de León, F., Herrmann, R., Feulner, P., Tiedemann, R., and Plath, M. (2008). Toxic hydrogen sulfide and dark caves: phenotypic and genetic divergence across two abiotic environmental gradients in *Poecilia mexicana*. *Evolution*, 62(10):2643–2659.
- Todd, P. M. and Miller, G. F. (1997). Biodiversity through sexual selection. *Artificial Life Five*, 5:289.
- Tomko, N., Harvey, I., Philippides, A., and Virgo, N. (2011). Many hands make light work: Group evolution and the emergent division of labour. In *Proceedings of the European Conference on Artificial Life*, pages 805–812, Paris, France. MIT Press.
- Vonlanthen, P., Bittner, D., Hudson, A. G., Young, K. A., Müller, R., Lundsgaard-Hansen, B., Roy, D., Di Piazza, S., Lurgiader, C. R., and Seehausen, O. (2012). Eutrophication causes speciation reversal in whitefish adaptive radiations. *Nature*, 482(7385):357–363.
- Whitley, D., Rana, S., and Heckendorn, R. B. (1999). The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology*, 7:33–48.
- Yaeger, L. (1994). Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or PolyWorld: Life in a new context. In *Artificial Life III*, pages 263–298. Addison-Wesley.

Collective Dynamics

Papers

Contextual Geometric Structures: modeling the fundamental components of cultural behavior

Bradly Alicea¹

¹Department of Animal Science, Michigan State University, East Lansing, MI 48823
bradly.alicea@ieee.org

Abstract

The structural complexity of culture cannot be characterized by simply modeling cultural beliefs or inherited ideas. Formal computational and algorithmic models of culture have focused on the inheritance of discrete cultural units, which can be hard to define and map to practical contexts. In cultural anthropology, research involving structuralist and post-structuralist perspectives have helped us better understand culturally-dependent classification systems and oppositional phenomena (e.g. light-dark, hot-cold, good-evil). Contemporary research in cognitive neuroscience suggests that complementary sets may be represented dynamically in the brain, but no model for the evolution of these sets has of yet been proposed. To fill this void, a method for simulating cultural or other highly symbolic behaviors called contextual geometric structures will be introduced. The contextual geometric structures approach is based on a hybrid model that approximates both individual/group cultural practice and a fluctuating environment. The hybrid model consists of two components. The first is a set of discrete automata with a soft classificatory structure. These automata are then embedded in a Lagrangian-inspired particle simulation that defines phase space relations and environmental inputs. The concept of conditional features and equations related to diversity, learning, and forgetting are used to approximate the goal-directed and open-ended features of cultural-related emergent behavior. This allows cultural patterns to be approximated in the context of both stochastic and deterministic evolutionary dynamics. This model can yield important information about multiple structures and social relationships, in addition to phenomena related to sensory function and higher-order cognition observed in neural systems.

Introduction

Why is cultural change so complicated? Intuitively speaking, it seems as though cultural change should be easy to predict. Given the adaptable nature of culture, changes in the environment should be quickly matched by corresponding changes in cultural representations. However, the need for cultural change often does not result in an adaptive response. In some cases, culture often seems to be maladaptive in the face of adaptive pressures. These anecdotal observations demonstrate that cultural change is highly complex. How can we represent this complexity using a computational framework? The patterns that define cultural behaviors across generations and contexts are most likely created via emergent and evolutionary processes. Unlike goal-

directed-behaviors such as reaching for a cup of water or following a scent, there is often no clear outcome to pursue. Cultural representations should “make sense” of procedural knowledge in a way that is not only flexible but also constrained by conceptual interlinkage.

Cultural systems have been understood using a number of theoretical perspectives. Structural (Levi-Strauss, 1969) and post-structural (Murdoch, 2006) perspectives are based on the notion that cultural life is based on a set of structures orthogonal to human cognition. These structures ostensibly emerge from common patterns of behavior over multiple generations, and represent the outcomes of cultural evolution. One signature of these ephemeral structures is the cognitive representation of oppositional sets, which are bounded by extreme concepts for each category. For example, there may be a phenomenological and objective category shared across cultures bounded by maximal luminance (light) and absolute lack of luminance (dark). The extremes of this category are bounded by human perceptual abilities, so that experience of each culture can be contained within.

A “structure” can be defined as sets of relationships between objects in the environment, or experiences that can vary from person to person but are grounded in the same underlying concepts. These structures, which are a critical and implicit component of human cultural practice, have an underappreciated computational potential. This is particularly useful since many of these features are essential to understanding the evolution of culture across multiple generations (Bourdieu, 1977). Even more importantly, these structures might be an essential feature of how cultural practices are represented in a neural architecture. In recent years, brain scientists have applied this idea to a system of oppositional sets called complementary pairs (Kelso and Engstrom, 2006). In this approach, oppositional sets are contingent upon coupling, oscillatory, and heterogeneity in the dynamics of neural circuits. While these approaches hold much promise for the study of culture and symbolic systems, there remains a need to more fully integrate dynamical and structural approaches. I propose that by combining the structural features of cultural practice with a quasi-evolutionary perspective will result in a model of cultural evolution that maps to both social phenomenology and physiological function.

In addition, cultural and symbolic behavioral systems share many features with physical systems that exhibit chaotic behavior. It is this combination of quasi-evolutionary and chaotic dynamics that makes my approach unique. The approach presented here, called **Contextual Geometric Structures (CGS)**, is a Lagrangian-inspired approach that focuses on the structural complexity of cultural and other symbolic behavioral phenomena. In this paper, I will introduce a **hybrid soft classification/hydrodynamics model** in the context of cultural phenomena. Initially, basic features of the contextual geometric structure model will be introduced. It will then be demonstrate how this model fits into the milieu of cultural diversity and evolution. This includes features that approximate complex and diverse phenomena. Finally, we will consider this model in the context of neuronal processes.

Contextual Geometric Structures

Prior approaches to modeling culture have included forays into population genetics and game theory (Boyd and Richerson, 1985; Cavalli-Sforza and Feldman, 1981; McElreath and Boyd, 2007), memetic representations (Hart, Krasnogor, and Smith, 2005; Goh, Ong, and Tan, 2009), specialized genetic algorithms (Reynolds and Peng, 2004; Gessler, 2010), and conceptual blending models (Coulson and Oakley, 2000; Grady, 2000). In this paper, a computational approach focusing on the structural complexity of culture will be introduced. While the CGS approach incorporates some elements of these prior approaches, this is a fundamentally new approach to the problem.

Contextual geometric structures provide advantages that previous models do not. Models inspired by population genetics and game theory are explicitly discrete and focus on inheritance, and so do not produce many of the nonlinear behaviors that culture embodies. While memetic and conceptual blending models may provide insights into the combinatorial potential of cultural change, neither are explicitly dynamical. While computationally efficient, specialized genetic algorithms do not express the fluid output of cultural behaviors not explicitly associated with beliefs. Perhaps greatest advantage of this approach is the mapping of both these properties to a set of formal, computable structures.

Model Components

The CGS approach consists of a hybrid model: a “soft” computational structure representing the individual automata and a dynamical system representing the environment. Each automaton represents an individual with a brain that houses multiple conceptual spaces we call kernels. The automata then interact in a flow field. The dynamics of this flow field reinforce evolutionary behaviors and complex structural patterns.

Single automata

The cultural repertoire of each automaton (or particle) uses a soft classification scheme to represent the elements of culture. **Soft classification** (Miotra and Hayashi, 2006), a fuzzy logic-inspired methodology, provides several advantages. One of these advantages involves the capacity to represent different cultural contexts in the same model. Another advantage involves the capacity to represent degrees of specific cultural and symbolic behaviors rather than merely its presence or absence.

All natural phenomena classified by any single cultural group has a membership function on a membership kernel (Figure 1), bounded by the capacity of a sensory system. The resulting cultural representation of a phenomena will sit somewhere on this scale. Unlike probabilistic or likelihood models, soft classification does not require related objects and categories to be transitive, distributive, or symmetrical. This allows for the generation of context, which is central to many existing theories of culture.

n -dimensional “Soft” Kernels

Figure 1 shows one- and two-dimensional examples of cultural representations of “hot” to “cold”. Figure 2 demonstrates the membership kernel for three different cultures. The logical structure consists of various **membership kernels** which serve to classify the experience of each automaton into a common, objective scale. This graded scale acts to link together related concepts as shown in Figure 1. In this sense, they can be high-dimensional structures. One- and two-dimensional structures tend to represent concepts related to practice, while higher-dimensional structures represent a mapping from neurobiology to the cultural domain (see equations 1-5).

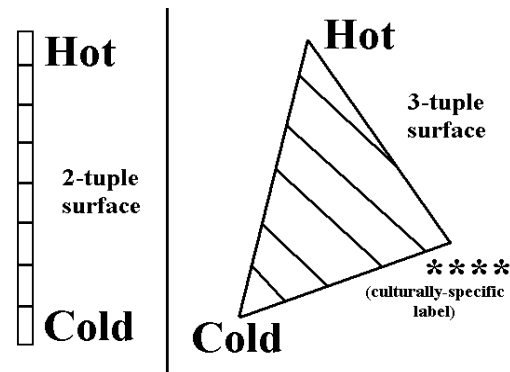


Figure 1. One- and Two-dimensional kernels embedded with n -tuple encodings.

In Figure 2, the objective scale for hot and cold stimuli has been mapped to a 2-tuple surface for three cultures (A-C) and their overlap. There will be **variability between individuals and cultures**, which can be evaluated using a common scale. To map physiological function to cultural and symbolic representations, **contextual anchors** will be used (see 3-

tuple surface, Figures 1 and 2). In context, contextual anchors provide a means to mediate the membership between hot and cold with procedural knowledge.

When different cultural categories overlap, it may be indicative of **previous contact**. However, separation between categories may also be indicative of cultural diversity in the form of **distinction**. Cultural distinction is a common feature of cultural evolution which can sometimes be imposed by its practitioners. In our context, we will assume that cultural distinction is an emergent feature, and is specified by the segregation factor (see Equation 6). Segregation or distinction is characterized by the non-overlapping region between B and C in Figure 2.

Environment

The environmental component of contextual geometric structures involves a second-order Lagrangian system with dynamics that produce solutions analogous to **Lagrangian Coherent Structures** (LCS - Mitra and Hayashi, 2006). LCS structures are defined as “ridges” of particles that aggregate in different portions of the flow field. Quantitatively, comparisons between particle positions can be made using either the **Finite Time Lyapunov Exponent** (FTLE - solved with regard to temporal divergence) or the **Finite Space Lyapunov Exponent** (FSLE - solved with regard to spatial divergence) (Haller, 2007; Lipinski and Mohseni, 2010). Characterization of these features can be encapsulated in a measure called the **iterated temporal divergence** (see Equation 7). This methodology has previously been applied as a generalized analogy for evolvability in biological evolution (Alicea, 2011). This work is an extension of this application, the schematic of which is shown in Figure 3.

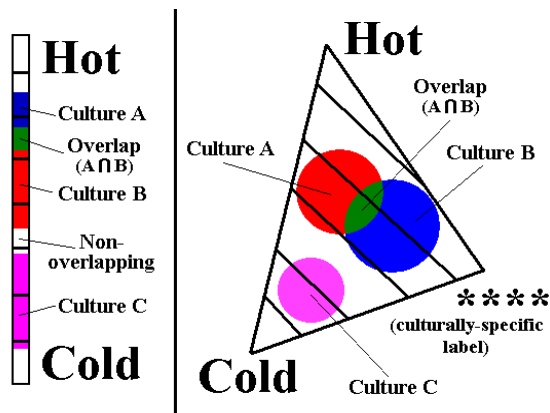


Figure 2. A soft classification kernel populated with the space for three different cultures. In this case, the same automaton is a carrier for three sets of cultural knowledge simultaneously.

As can be seen in Figure 3, the automata are initialized in the same location and then get diffused by the force field environment. The automata also have properties of **replicator vehicles** that reproduce according to specified parameters. While the selective

component of the model has yet to be specified completely, LCS-like models should produce outcomes dominated by evolutionary neutrality (Reidys and Stadler, 2001). In addition, our goal is to observe cultural diversity, which involves far-from-equilibrium and sub-optimal behaviors obscured by strong selective pressures.

When applied to cultural systems, the LCS approach (Tew Kai, et.al, 2009) typically involves **observing the diffusion of particles in a hydrodynamic force field and tracking the structures** that result (Figure 3). These structures are observed to collide, pull apart, and intermingle over time. Yet external forces introduced by the flow field can influence diffusion, and so the particles will still aggregate into recognizable and orderly structures. Contextual geometric structures show form as a consequence of evolutionary constraints and interactions between agents over time.

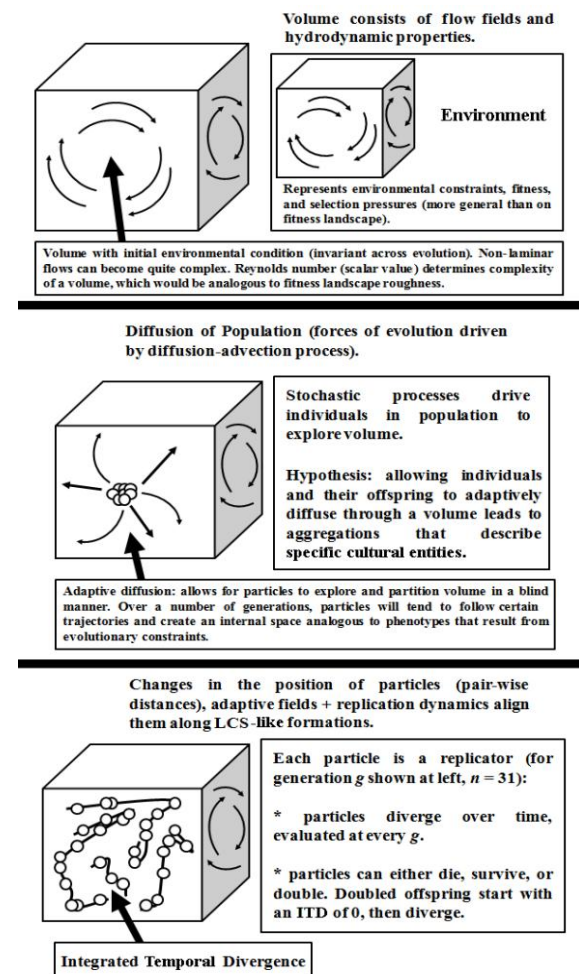


Figure 3. Cartoon depicting a typical contextual geometric structure simulation over the course of cultural evolution. TOP: initial condition, MIDDLE: active diffusion of the automata population, BOTTOM: final volume features contextual geometric structures.

Structures, Diversity, and Evolution

In order to better understand the role of evolution in the emergence of contextual geometric structures, it is important to take a closer look at the outcome of interactions between three distinct automata populations. Figure 4 shows an example run using automata from three distinct cultures (red, blue, and black). This 2-D LCS volume features 165 automata present at the following frequencies: black (0.35), blue (0.35), and red = (0.30). This allows us to observe a number of purely physical outcomes after the evolution of an initial population. The first of these are loosely-organized vortices, which can either be homogeneous (all automata of the same color) or heterogeneous (automata of multiple colors). The second physical feature is a cluster often found along edges of the volume. These aggregates can be either homogeneous or heterogeneous, and can be considered products of pure diffusion. The third physical feature is a ridge, which can be either homogeneous or heterogeneous and often leads to the formation of vortices. The fourth physical feature is a vortex, which is a tightly packed aggregation of automata which is usually homogeneous.

Yet how exactly do these formations map to the evolution of culture? Using a mixed initial population can lead to competition, selection, and other quasi-evolutionary dynamics. The soft classifications inherent to each automaton must be coordinated using a series of features based on principles of attraction and repulsion to allow the diffusion of automata within a flow field to exhibit behaviors relevant to cultural structures and practice. Three features are expected to produce a broad range of highly-complex and realistic cultural scenarios.

Initial condition of model

The choice of a hybrid soft classificatory/hydrodynamics model may allow us to observe evolution enforced by self-organization. The tracking of particle populations allows for complex dynamics to emerge out of interactions between automata and the environment. In the model presented here, a forcing mechanism more complex than uniform diffusion may be required to produce quasi-evolutionary dynamics (see [Supplementary Information](#)). I propose the use of virtual flow jets (embodied in rulesets), which can mimic the uniform diffusive properties of neutral evolution (Olca, Pottebaum, and Krueger, 2010). Likewise, we can approximate natural selection by adding $1/f$ noise to the flow field. This and other forms of asymmetric perturbation can mimic the directional properties of selection (Shlesinger, West, and Klafter, 1987).

Depending on force parameters that constrain the simulation environment, the simulation can yield vastly different behaviors. Yet the relational structure between concepts can remain quite similar across contexts. One feature of evolutionary systems is that they are often constrained to a particular evolutionary

trajectory by past trajectories and current features (Schwenk, 1995). These constraints combined with environmental fluctuations simulated by the addition of systematic noise produce quasi-evolutionary dynamics.

Features that shape evolution

As previously mentioned, systematic noise can be used to perturb the flow field. This perturbation can approximate different evolutionary dynamics. In a like manner, **conditional features** are top-down, deterministic perturbations of the flow field that act like selective mechanisms. Three conditional features are proposed: purity, associativity, and syncretism. These features are predicted to produce a wide range of contextual geometric structures that may be identified as complex cultural dynamics (see Figure 5). Each conditional feature operates on the n -dimensional kernels of each automaton. While a lack of selection can produce evolutionary dynamics, higher-level organizational features can also increase the adaptive capacity of an evolutionary system (Wagner, 2005; Dorigo, and Stutzle, 2004). In our system, this is realized via simple interaction rules which lead to complex and highly-ordered outcomes.

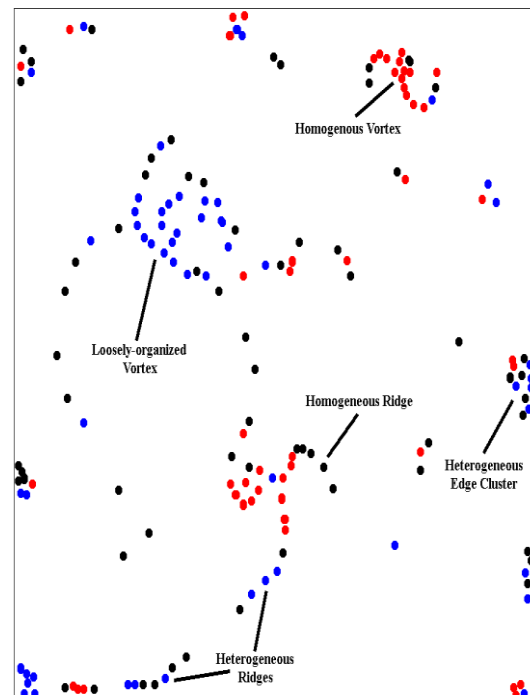


Figure 4. A 2-dimensional space representing an evolved population of automata representing three distinct cultures (Black, 58 automata; Blue, 58 automata; Red, 49 automata). Each subpopulation has a multifaceted set of relationships with regard to the other two.

Purity is successfully enforced when two or more distinct structures are formed. These structures are distinct in that all automata flow inward towards discrete vortices (Figure 5, Scenario #1). Over time,

automata of different subpopulations exhibit total separation from one another. Associativity is successfully enforced when automata flow outward from established vortices along several trajectories towards one another (Figure 5, Scenario #2). Associativity often results in heterogeneous structures, and may lead to interactions between subpopulations.

The effectiveness of the purity and associativity sorting mechanisms can be detected using the **conditional diversity measure**, shown in Equation 8. This measure provides a profile of all automata within a certain level of Lagrangian divergence in the flow field by using a single parameter D . When the value converges upon 0.5, the collection of automata that compose a loosely-associated structure or ridge is highly homogeneous. When the value approaches 0.0, the collection of automata is highly heterogeneous.

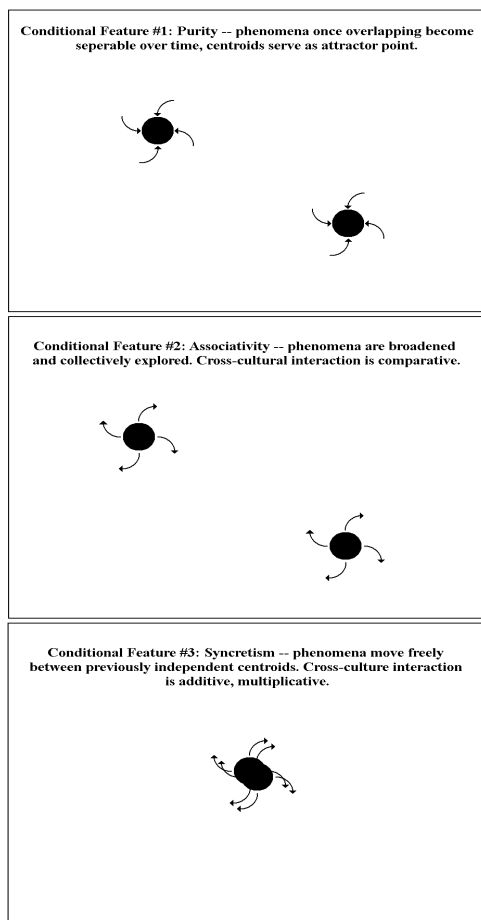


Figure 5. Three types of enforcing selection (conditional features) for the evolution of contextual geometric structures. Cartoon illustrates the general shape and mode of action characterized by each flow field modification.

Syncretism involves the dispersion of automata towards automata of a competing population. This generally involves automata that are aggregated around two or more vortices. Based on this conditional

feature, automata spiral outward from these aggregation centers towards each other in overlapping patterns (Figure 5, Scenario #3). The particles (automata) are freely interchanged in the resulting vortex and trailing flow (Figure 5, Scenario #3).

The predicted features shown in Figure 4 are approximations of what could be referred to as cultural practice space. In this sense, structures represent the aggregation of different cultures, which are distinct from individual automata holding representations for multiple cultures. This may allow us to make complex cross-cultural comparisons.

Intermittent and transient dynamics

A main assumption of this model is that variation in a flow field of variable turbulence might contribute to local changes in the rate of evolution. Indeed, actively manipulating the flow parameters is another way to observe the “churn” of cultural evolution. Yet the relationship between the two model components might also allow us to observe selective conservation across cultural structures and practices.

What is the evolutionary relationship between the kernel values housed by individual automata and the Lagrangian unfolding in environmental space? To address this, we constructed a rate measure for learning and forgetting (see Equation 9). This measure bridges the gap between model components by tying kernel value segregation between populations to their distance in the Lagrangian flow field. These distances between concepts of practice and the evolutionary trajectory of individual automata (respectively) can be thought of as gaps that are translated between the two models. Learning occurs in cases where the gap between kernel values for different populations of automata is transferred to the evolutionary space (e.g. where the ITD value becomes larger over time). Forgetting occurs when the gap between kernel values for different populations of automata is transferred to the evolutionary space (e.g. where the ITD value becomes smaller over time).

Applying this measure when comparing subpopulations refines the model’s ability to simulate the navigation of culturally-specific structures, which result in more coherent structures and life-like behavior. When very large r_{LF} values occur, learning predominates. When very small r_{LF} values occur, forgetting predominates. As in real culture, we expect representations of practice to fluctuate between extremes when the environment is unpredictable. In this model, such dynamics could be realized by simulating a turbulence regime (see [Supplementary Information](#)).

Conclusions

In this paper, I have proposed both an architecture and set of testable predictions for a model of cultural evolution focused on approximating the structures of practice. There are also several conclusions regarding the applicability of this model to real-world

settings. The ultimate goal is to model the diversity and evolutionary dynamics of context. The common features and shortcomings of this model can tell us something about the cultural structures related to practice.

Why choose this particular model? The soft classificatory structures were chosen as a way to map cultural practices to **both** a quantitative scheme **and** perceptual mechanisms in the brain. The fuzziness of this model is particularly useful in capturing the nuance that cultural representations tend to exhibit. Coupling this to a LCS-inspired model is done to extend the static nature of the classification scheme to an evolutionary context. It is my contention (see Alicea, 2011) that LCS-inspired models capture evolutionary phenomena that fitness landscapes cannot. In the model presented here, flow fields can help us better understand the dynamics of intermingling during cultural contact and intentional segregation based on cultural content. This can lead us to better theories about cultural universals and perhaps even the neural bases of culture.

The take-home message from this work is twofold. One part of the message is that the inability of culture to adapt to rapidly-changing environments is not simply inertia. The other part of this message is to suggest that the ability of culture to adapt rapidly to environmental challenges is not free of constraints. Given these conclusions, this method is not meant to be a general-purpose model for understanding every cultural phenomenon. Rather, the focus is on cultural practices and the structures that underlie descriptive structures.

To better understand the adaptive capacity of cultural systems, our ultimate goal is to characterize the labyrinthine features of a practice or ritual. This might explain why some practices are resistant to change (such as religious rites), while others can be highly improvisational (such as a jazz score). Notably, this model does not account for hierarchical and ecological relationships between cultural and social groups. Our focus is more on the origins of cultural complexity and the spontaneous nature of cross-cultural interplay.

Idiosyncrasies observed in the adaptive capacity of culture can be seen in behaviors unique to our approach. The [supplementary information](#) section provides a link to an Animation that demonstrates how automata and even entire structures can exhibit recursive behaviors such as local cycling and clustering by automata type. These are essential ingredients for determining cultural context, but need further development.

One key advantage of this model over previous approaches to modeling culture is its relevance to neurobiological processes. Objective categories that incorporate information about cultural context can be placed explicitly in the context of integrative mechanisms in the brain. Similar to a typical model of brain function, the fine-grained biological details are implicit in our soft classification model. Yet unlike a typical model of brain function, the evolution of collective behavior and shared cultural information over time are simulated using a physics-based model.

One example of dynamic, nonlinear neuronal processing related to symbolic behavior is multisensory integration. Multisensory integration involves the integration of visual, auditory, and somatosensory information at selective sites in the brain (Meredith and Stein, 1986). In mammals, the superior colliculus integrates visual and auditory sensory information for further processing relevant to the orienting function of attention (Macaluso, Frith, and Driver, 2000). This combination of senses is not linear, and the coincidence of stimuli in space and time results in a superadditive electrophysiological response (Holmes and Spence, 2005).

However, neural integration may not be limited solely to combining information from sensory systems (Goldman, Compere, and Wang, 2007). In this model, the soft classification schemes form the basis of cultural practice structures as they might be represented in the brain. For example, a group membership ritual or political campaign can involve many procedures, classifications, and judgements about the natural world that make no sense in isolation or outside the context of a specific ritual. As a neural mechanism, integration may also play a critical role in switching between the logic of cultural structures and active cognition, and may be particularly important when approximating diverse responses to common stimuli that due to context.

Future work should also focus on several common phenomena in cultural systems. One example of this is when selected dimensions of a kernel (such as the light-dark or good-bad oppositions) are treated as the entire practice. This often occurs in fundamentalist religions. Another target for future research involves understanding seemingly illogical behaviors, such as reinforced ritualized behaviors, despite the need for cultural change. Placing the evolution and information processing of these phenomena within a logical framework may lead to further advances in understanding behavior and ultimately human nature.

Supplementary Information

Please visit <http://syntheticdaisies.blogspot.com/p/fluid-models-of-evolutionary-dynamics.html> for supplemental materials (graphs, animations, and practical examples).

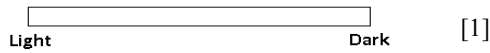
Methods and Equations

Particle structures. The number of potential structures that can interface with cognitive and neural processes can be quite large. We constructed five distinct particle structures, which can be defined as combination of dimensions representing both the fundamental limits of a neural subsystem (e.g. vision, touch, auditory, gustatory) and the centroid of a contextual variable (e.g. fluctuation, umami, modulation). The contextual

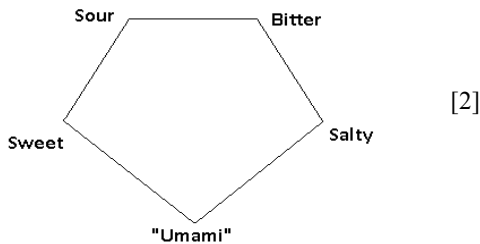
variable has cultural meaning, and site in relation to these perceptual limits.

Soft classification allows for an n -tuple representational scheme which is not mutually exclusive. Phenomena can belong to two or more categories simultaneously, differing only in terms of degree. For example, changes in “light” do not result in corresponding changes to the “dark” classification. The use of contextual anchors (which also employ soft classification schemes) concurrent with the neural mechanism dimensions allows for non-additive cultural representations that approximate the sub- and super-additivity common in neural mechanisms of sensory integration.

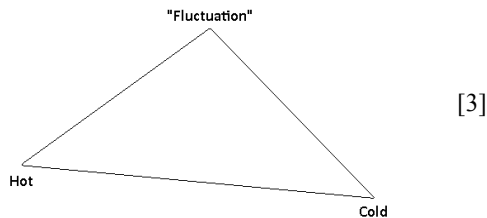
2-tuple without a contextual anchor. The first (and simplest) kernel design is the 2-tuple without a contextual anchor based on light sensing and visual perception. The example in [1] shows a binary opposition representing the transition between light and dark, an exemplar of which can be stated as [0.6, 0.2].



5-tuple with a contextual anchor. The second kernel design is a 5-tuple with a contextual anchor, and maps to the human gustatory system. The example in [2] shows a discrete set of tastes, an exemplar of which can be stated as [0.2, 0.2, 0.4, 0.8, 0.2].

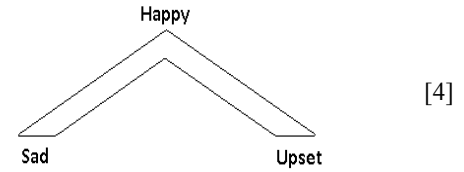


3-tuple with contextual anchor. The third kernel design is a 3-tuple with a contextual anchor, and maps to the function of thermoreceptors in the haptic system. The example in [3] shows a discrete set of tastes, an exemplar of which can be stated as [0.6, 0.2, 0.9].

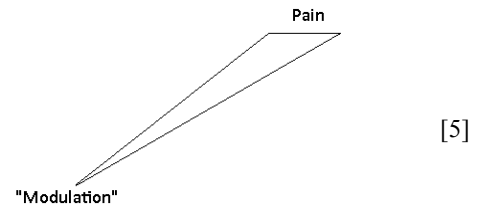


3-tuple without contextual anchor. The fourth kernel design is a 3-tuple without a contextual anchor, and

maps to the functions of arousal and emotion. The example in [4] shows a discrete set of emotional states, an exemplar of which can be stated as [0.1, 0.5, 0.3].



2-tuple with contextual anchor. The fifth kernel design is a 2-tuple with a contextual anchor, and maps to the function of nociceptors in human tissues. The example in [5] shows the degrees between the pain state and modulation of pain (a highly parallel process but represented here as a point), an exemplar of which can be stated as [0.6, 0.8].



Iterated Temporal Divergence (ITD). Iterated Temporal Divergence is defined using the following equation

$$L_t(X_0) = \int_t^{t+1} (v - v) | F_t^s(X_0) ds \quad [6]$$

where the divergence between two particles subject to the same flow field is integrated over a finite time period, $t: \rightarrow t + 1$.

Segregation Factor. The segregation factor is used to understand changes in the distribution of values for a particular soft classification kernel. Sets that define the structure of a certain cultural feature can become segregated over time, resulting from interactions with other particles in the flow field. This can be defined as

$$S = |\sum I_{ij}|, |\sum I_{ij}| > 0 \quad [7]$$

where a value of $S \rightarrow 1.0$ results in a maximization of movement towards discrete positions on the particle.

Conditional Diversity. To measure the distribution of automata within a given ridge or vortex, we can use a measure of conditional diversity. This measure provides us with a distribution of automata in the flow field for all automata within a certain value of the ITD measure (see equ. [6]). This measure can be stated as

$$D = \sigma(p_1, p_2, \dots, p_n)$$

$$p_i = \frac{A_i}{A_{tot}} \quad [8]$$

$$A_i = \underset{X_0}{\operatorname{argmax}} L_t X_0 \leq L_t X_0 \geq 0$$

where σ equals the variance of set p_n , A_i equals all automata for a specific subpopulation below the threshold value for the ITD measure, p_i is the number of automata in a specific subpopulation, A_{tot} is the total number of automata, and p_n is the number of subpopulations in the simulation.

Rate of learning and forgetting. To measure the relationship between the kernel representing the structure of practice and the Lagrangian model representing evolution, a rate can be used to characterize a cultural distance between populations based on distinctions in practice. The can be expressed as

$$r_{LF} = \frac{L_t(X_0)}{S_{pi} - S_{pj}} \quad [9]$$

where $L_t(X_0)$ is the iterated temporal divergence, and S_{pi} and S_{pj} are segregation factors for different automata populations housing a particular kernel.

References

- Alicea, B. (2011). Lagrangian Coherent Structures (LCS) may describe evolvable frontiers in natural populations. arXiv:1101.6071.
- Bourdieu, P. (1977). *Outline of a Theory of Practice*. Cambridge University Press, Cambridge, UK.
- Boyd, R. and Richerson, P.J. (1985). *Culture and the evolutionary process*. University of Chicago Press, Chicago, IL.
- Cavalli-Sforza, L.L. and Feldman, M.W. (1981). *Cultural transmission and evolution: a quantitative approach*. Princeton University Press, Princeton, NJ.
- Coulson, S. and Oakley, T. (2000). Blending basics. *Cognitive Linguistics*, 11(3-4), 175-196.
- Dorigo, M. and Stutzle, T. (2004). *Ant Colony Optimization*. MIT Press, Cambridge, MA.
- Gessler, N. (2010). Fostering Creative Emergence in Artificial Cultures. *Proceedings of Artificial Life XII*, 669-676.
- Goh, C-K., Ong, Y.S., and Tan, K.C. (2009). Multi-objective memetic algorithms. Springer, Berlin.
- Goldman, M., Compte, A., and Wang, X-J (2007). Neural Integrators: recurrent mechanisms and models. *New Encyclopedia of Neuroscience*.
- Grady, J. (2000). Cognitive mechanisms of conceptual integration. *Cognitive Linguistics*, 11(3-4), 335-346.
- Haller, G. (2007). Uncovering the Lagrangian Skeleton of Turbulence. *Physical Review Letters*, 98, 144502.
- Hart, W.E., Krasnogor, N., and Smith, W.E. (2005). *Recent advances in memetic algorithms*. Springer, Berlin.
- Holmes, N.P. and Spence, C. (2005). Multisensory integration: Space, time, and superadditivity. *Current Biology*, 15(18), R762-R764.
- Kelso, J.A.S. and Engstrom, D.A. (2006). *The Complementary Nature*. MIT Press, Cambridge, MA.
- Levi-Strauss, C. (1969). *The elementary structures of kinship*. Beacon Press, Boston.
- Lipinski, D. and Mohseni, K. (2010). A ridge tracking algorithm and error estimate for efficient computation of Lagrangian coherent structures. *Chaos*, 20, 017504.
- Macaluso, E., Frith, C.D., and Driver, J. (2000). Modulation of human visual cortex by crossmodal spatial attention. *Science*, 289, 1206-1208.
- McElreath, R. and Boyd, R. (2007). *Mathematical Models of Social Evolution: a guide for the perplexed*. University of Chicago Press, Chicago, IL.
- Meredith, M.A. and Stein, B.E. (1986). Visual, auditory, and somatosensory convergence on cells in superior colliculus results in multisensory integration. *Journal of Neurophysiology*, 56(3), 640-662.
- Mitra, S. and Hayashi, Y. (2006). Bioinformatics with Soft Computing. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 36(5), 616-635.
- Murdoch, J. (2006). *Post-structuralist geography: a guide to relational space*. Sage, Thousand Oaks, CA.
- Olcay, A.B., Pottebaum, T.S., and Krueger, P.S. (2010). Sensitivity of Lagrangian coherent structure identification to flow field resolution and random errors. *Chaos*, 20, 017506.
- Reidys, C. and Stadler, P.F. (2001). Neutrality in fitness landscapes. *Applied Mathematics and Computation*, 117(2-3), 321-350.
- Reynolds, R.G. and Peng, B. (2004). Cultural Algorithms: Modeling of How Cultures Learn to Solve Problems. *IEEE Proceedings on Tools with Artificial Intelligence*, 166-174.
- Schwenk, K. (1995). A utilitarian approach to evolutionary constraint. *Zoology* 98:251-262.
- Shlesinger, M.F., West, B.J., and Klafter, J. (1987). Levy dynamics of enhanced diffusion: Application to turbulence. *Physical Review Letters*, 58, 1100-1103.
- Tew Kai, E., Rossi, V., Sudre, J., Weimerskirch, H., Lopez, C., Hernandez-Garcia, E., Marsac, F., and Garcon, V. (2009). Top marine predators track Lagrangian coherent structures. *Proceedings of the National Academy of Sciences USA*, 106(20), 8245-8250.
- Wagner, A. (2005). *Robustness and Evolvability in Living Systems*. Princeton University Press, Princeton, NJ.

Effects of Local Communication and Topology on Collective Movement Initiation

Brent E. Eskridge

Department of Computer Science and Network Engineering, Southern Nazarene University, Bethany, OK 73008
beskridge@snu.edu

Abstract

Collective movement in autonomous systems, such as a team of robots, are frequently implemented using complex interaction rules and have significant communication requirements. These restrictions frequently relegate such systems to static, simplified environments. In contrast, collective movements in natural systems consistently occur in dynamic, complex environments in which significant communication is either impractical or impossible, and have been successfully modeled using simple, local interaction rules. In the work presented here, one such model is extended to include local communication and the spatial distribution of the group so that it can eventually be used as a guide for developing artificial systems capable of cohesive, collective movements. The extended model predicts that a reliance on local communication does not necessarily mean there will be a significant loss in the expected success of collective movement attempts if appropriate interaction rules are chosen. Furthermore, the model predicts that the addition of local communication, in conjunction with the topology of the group, results in higher expected success in attempting collective movements for individuals with central locations in the group as compared to individuals occupying edge locations.

Introduction

Collective movement is a necessary consequence of living and working in groups. As a result, considerable attention has been paid to the performance benefits of effective collective movements in both artificial and natural systems. However, there has historically been a dichotomy in how artificial and natural systems arrive at collective movement. Artificial systems, such as **multi-robot systems (MRSs)**, generally interact using complex rules that require precise sensor information and significant communication between robots, and are limited to operations in controlled environments. On the other hand, collective movement in natural systems can be successfully modeled using simple, local interaction rules requiring little to no explicit communication between group members and operate in complex, dynamic environments. To achieve the adaptability and simplicity of natural systems, the design and development of artificial systems can take inspiration from natural systems, especially in the area of collective movement.

In MRSs, coordinated actions such as collective movements are generally achieved through explicit, global communication (Balch and Arkin, 1994; Ampatzis et al., 2008). While explicit, global communication can be straightforward to implement, there are significant problems with its application in MRSs. Not only is it sensitive to environmental conditions, but explicit communication has problems scaling to large teams of robots (Anderson and Papanikolopoulos, 2008). In the worst case, the computational complexity of coordinating n individuals using explicit, global communication is $O(n^2)$ (Klavins, 2003). Limitations in communication are also a factor in collective movements in natural systems. Although small groups exhibit explicit, global communication, its use in a group cannot reliably scale from the individual to the group (Couzin, 2009). As a result, large groups in nature use local communication and frequently rely on environmental cues, a form of implicit communication (Drapier et al., 2002).

Researchers have recently proposed a number of models for collective movement based on observations of natural systems (Jacobs et al., 2011; Pillot et al., 2011; Sueur et al., 2011; Sueur and Deneubourg, 2011). However, the majority of these cited models were developed based on observations of groups of less than 15 members and assume global communication. As a result, for a model to be useful where global communication is not an option, it must be amenable to the addition of local communication. The ease with which local communication can be added to a model is primarily determined by the complexity of the rules governing the interactions between individuals. If a model's rules are relatively simple, then adding local communication is reduced to the task of limiting the interaction rules currently governing a given individual, based on the individuals with which it is communicating. Group information, such as an individual's actions, is propagated throughout the group using the relationships among group members as determined by the group's spatial distribution, or the group's *topology*.

For this initial investigation, the model proposed by Gautrais (2010) was chosen for extension. While there are many candidate models, some of which already use local commu-

Parameter	Value
τ_o	1290
α_c	0.009
γ_c	2.0
ε_c	2.3
α_f	162.3
β_f	75.4

Table 1: These model parameters were determined through direct observation of collective movement attempts in white-faced capuchin monkeys (Petit et al., 2009; Gautrais, 2010).

nication, this model’s use of simple, yet effective interaction rules facilitated the addition of local communication. Furthermore, it is anticipated that the simplicity of this model will also facilitate its application to artificial systems, such as a team of robots. To evaluate the effects of local communication and topology on the model’s predictions, simulations were performed with two variations of the model for a range of group sizes. The results of these simulations show that the extended model predicts only a small reduction in the mean expected success percentage for collective movement attempts, if the appropriate definition of an individual’s local neighborhood is chosen. Furthermore, the extended model predicts that the addition of local communication and topology results in a comparatively higher mean expected success rate in collective movement initiations for individuals occupying central positions in the group, when compared to individuals occupying edge positions.

Collective Movement Model

The model chosen for extension was developed through observations of collective movement attempts in a group of ten white-faced capuchin monkeys (Petit et al., 2009; Gautrais, 2010), and was later confirmed in observations of sheep groups ranging in size from 2–8 members (Pillot et al., 2011). It uses three interaction rules to govern the decision-making process involved in starting collective movements. The first rule assumes that all individuals within the group can initiate a collective movement attempt with a rate of $1/\tau_o$ (see Table 1). While this assumption may not hold for groups with dominant leaders, studies have shown that it is a viable assumption for egalitarian animal groups, such as the capuchin monkeys used in the model’s development.

Since the model assumes global communication, once an individual initiates a collective movement, the remaining individuals are assumed to have observed the initiation attempt and have the opportunity to follow the initiator. The second rule describes the rate at which followers join the collective movement attempt and is calculated by $1/\tau_r$. The time con-

stant τ_r for the following rate is calculated by the following:

$$\tau_r = \alpha_f + \beta_f \frac{N - r}{r} \quad (1)$$

where α_f and β_f are constants determined through direct observation (see Table 1), N is the number of individuals in the group, and r is the number of individuals following the initiator (Petit et al., 2009; Gautrais, 2010). Note that as the number of individuals following the initiator increases, the rate at which individuals join the movement also increases.

Not all initiation attempts are successful as initiators often cancel and return to the group. The third rule calculates this cancellation rate by the following:

$$C_r = \frac{\alpha_c}{1 + (r/\gamma_c)^{\varepsilon_c}} \quad (2)$$

where α_c , γ_c , and ε_c are constants determined through direct observation (see Table 1), and r is the number of individuals following the initiator. Note that as the number of individuals following the initiator increases, the rate at which the initiator cancels an initiation decreases. Also, simulations of the model include the implicit assumption that a successful collective movement requires all of the members of the group to participate, since there is a non-zero probability of canceling even if all but one member participates. While this is not necessarily the case in nature, cohesive, collective movements are the primary objective of this work and, as such, incomplete movements are considered failures.

Fundamental to these rules is the concept of mimetism, in which an individual’s probability of choosing an action is related to the number of individuals already performing the action (Pyritz et al., 2011). A variety of types of mimetism have been observed in natural systems and are usually differentiated by an individual’s choice of *whom* to mimic. In this model, *anonymous mimetism*, or *allelomimetism*, is used since individuals do not use the identity of group members when choosing whom to mimic. Anonymous mimetism is particularly useful for groups in which membership frequently changes and information such as an individual’s reputation may not be available. While information regarding a specific robot’s capabilities could be used in determining whom to mimic in a MRS, an anonymous model is useful as it represents a worst-case scenario.

Extending the Model

Scaling up the number of individuals within the group presents a choice regarding how a larger group size affects the model. Since it was developed for a small group with global communication, the model assumes each individual directly observes and interacts with every other individual within the group. However, when the number of individuals is scaled up, this assumption may no longer hold since spatial and cognitive constraints can limit the number of neighbors with which an observer interacts. This primarily affects

the calculation of the following rate constant, τ_r (see Equation 1), which uses the size of the entire group, N .

One option to consider is that the model does not need modification and that the assumptions regarding global communication are correct, regardless of group size. Furthermore, since white-faced capuchin monkeys are commonly found in groups of less than 20 members, with few reaching 30 members, their behaviors may not *need* to work with large groups (Fragaszy et al., 2004). However, while global communication is the easiest to implement and is reasonable given past work, evidence from nature does not support its use in large groups. For example, Pillot et al. (2011) noted that crowding in a group prevents an individual from observing all but its closest neighbors.

If global communication is not an option, then the flow of information between individuals becomes an important factor in the success of a collective movement attempt. Since the number of individuals with which an observer interacts is limited, only the individuals observing an initiation attempt are capable of following. This presents a choice regarding the value of N in the calculation of the following rate constant, τ_r , from Equation 1. The first option is to use the size of the entire group, denoted G , so that $N = G$. While this results in the same following constant as with global communication, the number of individuals capable of following the initiator is now limited to the number of individuals observing the initiator. As a consequence, the odds of the initiator canceling increase since the number of potential followers is limited. This choice also includes an implicit assumption that, although the number of group members with which an individual interacts is limited, group members know the size of the group and know the state of all the other members of the group. As such, this choice represents a logical contradiction, but it is still a useful choice against which the predictions of other models can be compared.

An alternative to using the size of the entire group is to use the number of individuals with an individual directly interacts. Ballerini et al. (2008) have shown that starlings appear to interact with, on average, a fixed number of neighbors. If this is true for other groups, then the number of nearest neighbors, denoted N_c , could be used so that $N = N_c$. The following rate constant τ_r would then be independent of the group size, unlike the previous option of using $N = G$. Intuitively, this appears to be a better choice since individuals are unlikely to be capable of observing all the individuals within a large group and are frequently found to mimic their closest neighbors.

Restricting the group to local communication introduces other side effects into the model beyond simply limiting which individuals can observe an initiator. First, since group members can be unaware that a movement has been initiated, unaware individuals are free to initiate a movement of their own. As a result, multiple initiators can be present at any given time within the group and competing for follow-

ers. Furthermore, since a movement attempt is considered successful if all the individuals choose to depart, either as an initiator or a follower, then it is entirely possible for a successful collective movement to be comprised of multiple groups, each with its own initiator. While this may not result in the desired *cohesive*, collective movement, an investigation into multiple group movements is reserved for future work. Lastly, the potential presence of multiple initiators means that a movement attempt is only considered a failure if *all* the initiators cancel. As long as one initiator remains, there is potential for success.

Numerical Implementation

To evaluate the effects of scaling up the group size, numerical simulations were performed using three different models. The first was the original model that assumed global communication within the group and the group size for the following rate calculations (i.e., $N = G$). Since global communication was assumed, the topology of the group did not have any effect on the simulation. While, as previously mentioned, this option seemed unlikely in many cases, it did provide a baseline against which the other models could be compared. The second model assumed only local communication within the group, but still used the group size for following rate calculations (i.e., $N = G$). The last model also assumed only local communication, but used the number of directly interacting neighbors for following rate calculations (i.e., $N = N_c$). While empirical observations of natural systems consistently yield N_c values in the range 6–7 (Ballerini et al., 2008), a value of $N_c = 10$ was used for these simulations to remain consistent with the original model and minimize the number of confounding variables.

For each model, group sizes from 10 to 100 individuals were evaluated. For the local communication models, thirty different evaluations were performed for each group size, each with different initial conditions, namely, a different random seed and topology, since the topology of the group influenced the results. In each evaluation, individuals were assigned random locations in a two-dimensional plane within a distance of 10 of the origin. These locations were then used to determine the $N_c = 10$ nearest neighbors for each individual and, therefore, the topology of the evaluation. While there are other methods for building random networks, this approach was used since it is the one that will be used when higher fidelity simulations are performed involving movement of individuals in a two-dimensional environment. For all three models, a single evaluation consisted of 20,000 simulations, each constituting a single attempt at a collective movement. All individuals had approximately the same number of initiation attempts as the initiation rates for all individuals were the same. Furthermore, the following and cancellation rates were the same for every individual in the group with the only differences between individuals being their nearest neighbors, as determined by their locations

within the group. The model parameters used were the same as those used in the original model (see Table 1), which were determined through direct observation of collective movement attempts in white-faced capuchin monkeys (Petit et al., 2009; Gautrais, 2010).

To quantify the effects of scaling the group size and using the group's topology, a variety of metrics were used. The primary measures of success were the overall percentage of initiation attempts that proved to be successful, referred to as the *success frequency*, and the relative success of individual initiations. The relative success of an individual in initiating collective movements was calculated as follows:

$$\text{Leadership} = \frac{\text{Number of moves led by } L}{\text{Total number of moves}} \quad (3)$$

which is the same as previous work (Gautrais, 2010). To assess an individual's significance within the group due to its position in the topology, two different measures were used. The first measure was the eigenvector centrality of the individual, which is a common measure of significance used in social network analysis (Wey et al., 2008). It quantifies how closely the individual is connected to the other individuals within the group. It is especially useful for collective movements and in highly connected networks (Sueur and Petit, 2008; Kasper and Voelkl, 2009). The second measure used was based on the topology of the group and an individual's interacting, nearest neighbors. This measure, referred to as an individual's *mimicking neighbors*, was the total number of individuals for whom a given individual was one of $N_c = 10$ nearest neighbors. Using this measure, individuals that had a larger number of mimicking neighbors had more influence within the group than those with fewer mimicking neighbors, since they had the potential to transmit information throughout the group faster.

Results and Analysis

For a group size of 10, there was no statistically significant difference between any of the three models. This indicates that the modifications made to the model to accommodate local communication did not alter the model to the extent that it was unable to make the same predictions. However, as the group size was increased, the differences between the model predictions were readily apparent. Figure 1 shows the predicted success frequency versus group size for each model. As the group size was increased, the predicted success frequency for both the global communication model and the local communication model using $N = N_c$ increased until they both reached an asymptotic limit of slightly less than 0.5. On the other hand, the predicted success frequency for the local communication model using $N = G$ dropped as the group size was increased. While the model using global communication predicted higher success frequencies than both local communication models at a statistically significant level for group sizes larger than 25

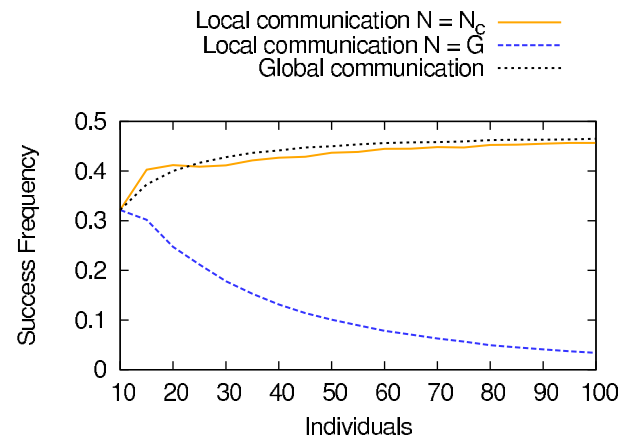


Figure 1: The frequency of successful collective movements as a function of group size for each treatment are shown. Confidence levels are omitted for clarity.

(Student's t-Test, $p \approx 0$), the practical difference between the global communication model and the local communication model using $N = N_c$ is less significant. Using Cohen's d statistic to determine the effect size between the two models, the largest predicted effect size was $d = 2.75$ for a group size of 45. Although this is traditionally considered a large effect size, the combination of 30 evaluations and a group size of 45 produced a large number of samples for each model and resulted in a small standard deviation. In practical terms, the use of the local communications model using $N = N_c$ resulted in at most a predicted loss in success frequency of 4.17% over all group sizes. On the other hand, the local communication model using $N = G$ resulted in a predicted loss in success frequency of 93.7% for a group size of 100. Given these predictions, and the fact that collective movements of thousands of individuals are frequently observed in nature, the remainder of the reported results are for the local communication model for which $N = N_c$.

Figure 2 shows two measures of an individual's significance within the group, namely the eigenvector centrality and the number of mimicking neighbors, vs. the expected leadership success (see Equation 3) of the individual for group sizes of $G = 20, 60$, and 100. A blue line denotes the line of best-fit for each set of data. For both measures, there was a clear correlation between the significance of the individual within the group and its expected leadership success in each group size. Although the mean expected leadership success of individuals decreased as the number of individuals increased, this was to be anticipated since there were more individuals capable of initiating movements.

The Pearson product-moment correlations between the expected leadership success of an individual and each of the two significance measures are shown in Figure 3. Each correlation was statistically significant with $p \approx 0$. Correlations between the number of mimicking neighbors and the ex-

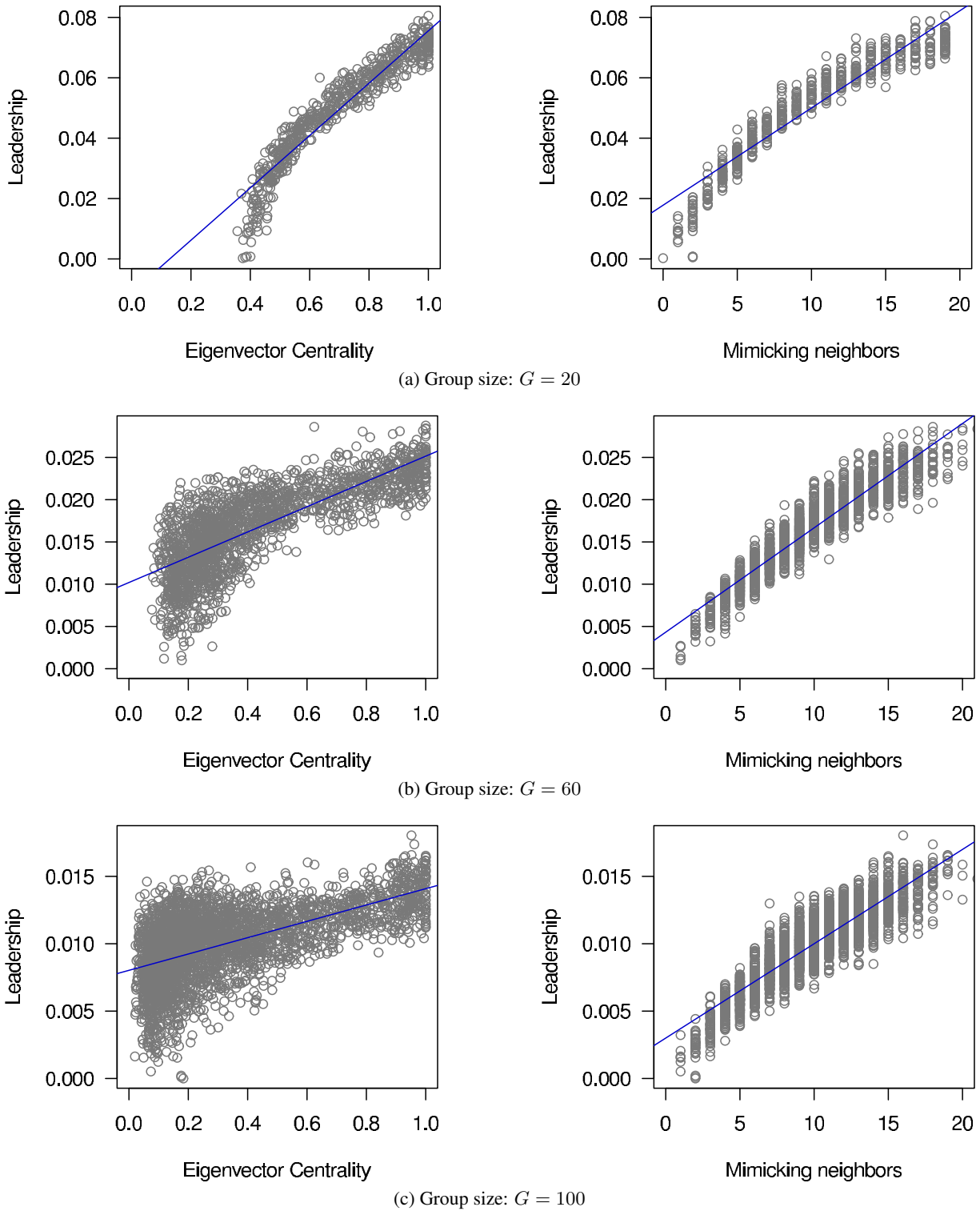


Figure 2: The two measures of an individual's significance within the group, namely the eigenvector centrality and the number of mimicking neighbors, vs. the expected leadership success of the individual for the local communication model using $N = N_c$ are shown. Each circle represents an individual from a single evaluation. Blue lines indicate the line of best-fit.

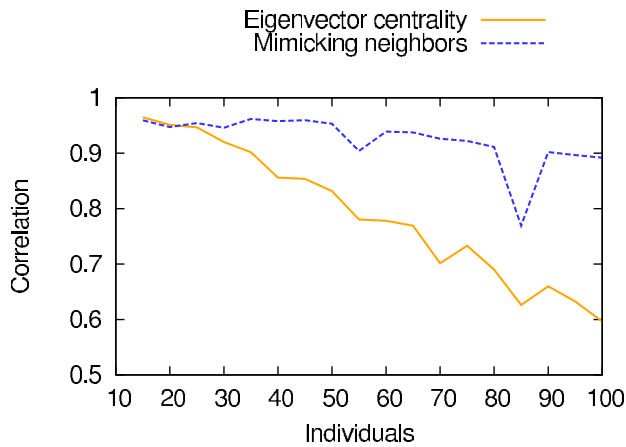


Figure 3: The correlation between two measures of an individual's significance within the group, namely the eigenvector centrality and the number of mimicking neighbors, and the expected leadership success of the individual are shown. These results only represent simulations for the local communication model in which $N = N_c$. Confidence levels are omitted for clarity.

pected leadership success was stronger than the correlations for the eigenvector centrality for large group sizes. While these linear correlations are statistically significant, non-linearities are visible for low measures of significance. For example, the leadership versus eigenvector centralities of approximately 0.5 and larger, while a non-linear correlation is visible for values less than 0.5. These non-linearities are present in the results for each of the group sizes evaluated. The loss of correlation as the group size increased could be attributed to the fact that, with a larger group size, outliers are more likely to be present within the group. One study saw a similar increase in noise as the group size was scaled up, even though the groups sizes were small compared to these simulations (Pillot et al., 2011).

Figure 4 illustrates this correlation between the number of mimicking neighbors and the expected leadership success using the topology for a run using a group size of 45 and the local communication model in which $N = N_c$. In this figure, the size of the individual represents the number of mimicking neighbors with larger sizes denoting more mimicking neighbors. The color denotes the individual's expected leadership success (see Equation 3), with orange denoting low success and blue denoting high success. While the individuals centrally located within the group have higher expected leadership success, it is not their location *per se* that correlates with their success. Rather, their expected leadership success is correlated with the number of mimicking individuals, which is a byproduct of their location within and the distribution of the group.

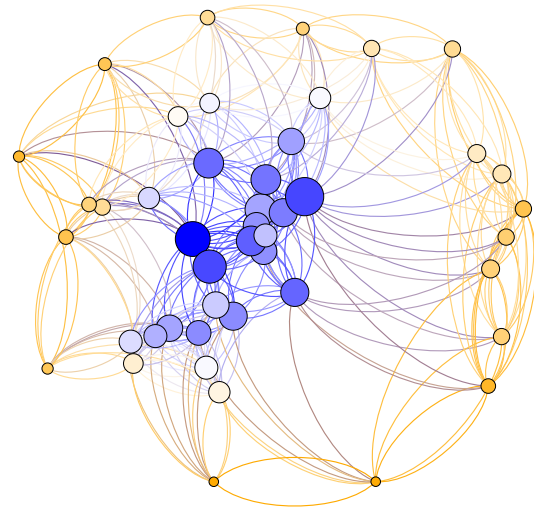


Figure 4: The group topology for a simulation using a group size of 45 is shown. An individual's color denotes its expected leadership success and ranges from orange, denoting low success, to blue, denoting high success. The size of the individual represents the number of mimicking neighbors with larger sizes denoting more mimicking neighbors.

While there was a noticeable drop in the correlation for the number of mimicking neighbors for a group size of 85, further analysis revealed that the topology from a single evaluation with a tightly-knit group of individuals produced outlier results (see Figure 5). Performing the correlation analysis with Spearman's rank correlation, which is less sensitive to outliers, resulted in a correlation of 0.876, which is more consistent with the trend observed in Figure 3.

Discussion

There are a number of conclusions that can be drawn from these predictions. First, the restriction of a group to local communication results in a minimal drop in the mean expected probability of success for collective movement attempts, depending on the interaction rules used. This is significant as it means that the requirement for global communication that is present in many models can be removed with only a small drop in performance, given the appropriate environment and inter-individual interactions. The model in which $N = N_c = 10$ predicts that the mean expected probability of a successful collective movement would be only slightly less than the model using global communication. On the other hand, the local communication model in which $N = G$ predicts that the probability of success would drop to less than 5% as the number of group members approaches 100. As was previously discussed, the difference in these predictions is due to the rate at which individuals follow the initiator. In the model using $N = G$, individuals base their decision-making on the actions of the entire

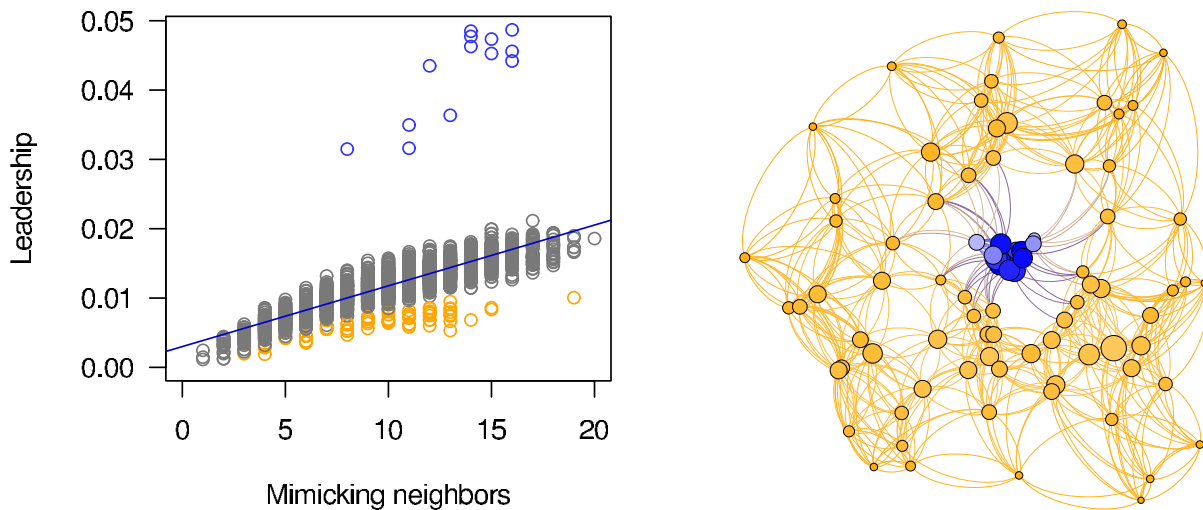


Figure 5: The number of mimicking neighbors vs. the leadership success of individuals in all simulations using a group size of 85 and $N = N_c$ is shown on the left. Individuals from a single run representing outlier results are emphasized in color. Individuals in blue had high leadership success and individuals in orange had lower success. The topology of the individuals in the simulation is shown on the right with the same coloring scheme. The size of an individual correlates to the number of mimicking neighbors with larger individuals having more.

group, even though the number of group members which they observe and with which they interact is limited. As was noted earlier, this presents a logical contradiction in the model, the result of which is a loss in predicted frequency of successful collective movement initiations.

Second, the addition of local communication and topology to the collective movement model resulted in statistically significant correlations between the centrality of an individual within the group and the individual's leadership success. While there are other studies that have shown that individuals in a central location within a group of capuchins are more successful leaders than those occupying edge positions (Leca et al., 2003), this is the first known work to demonstrate that a model using local communication and topology is sufficient to predict those results. Although social considerations may determine the spatial location and distribution of a group (Bode et al., 2012), both local communication models use anonymous mimetism. Therefore, one can conclude that this model predicts that the use of local communication in a spatial distribution of individuals is sufficient to produce consistent leadership by centrally located individuals, even in the presence of anonymous mimetism. However, these models assume identical cancellation rates for all members of the group, which may not always be the case.

Lastly, it should be noted that these results rely on the constraint that the topology, and, therefore, the communication network, was fixed throughout an entire simulation. Since there was no actual movement involved in the simulation, the neighbors with which an individual interacts re-

mains constant, regardless of their participation in a collective movement. Future work will determine whether the effects of removing this constraint.

Conclusions and Future Work

Effective, cohesive collective movements provide a variety of benefits for group members. Although there has historically been a difference between how these movements are modeled in natural and artificial systems, there are compelling motivations to use models derived from collective movements in natural systems to inform the design of artificial systems. To that end, the work presented here has extended a simple model of collective movement to operate effectively with large groups and serve as a guide for developing artificial systems capable of cohesive, collective movements. To accommodate larger groups using realistic physical constraints, the model was modified to use local communication and the topology of the group. Based on simulations with varying group sizes, the extended model predicts that restricting the group to only local communication can result in only a slight drop in the predicted success frequency of collective movements, if the appropriate interaction rules are used. In particular, the local communication model in which individuals base their decisions on the actions of their nearest neighbors predicts only a minimal loss in success frequency, while the model in which individuals base their decisions on the actions of the entire group predicts that collective movement initiations would rarely, if ever, succeed. In addition to these success frequency predictions, the extended model predicts that the combination

of local communication and topology results in significantly higher expected leadership success for individuals that are centrally located within the group as compared to individuals located at the edge of the group.

This work represents the initial stages of research into promoting emergent leadership and cohesive, collective movements in robot teams and there are a variety of opportunities for future work. First, the motivations of individuals to initiate a movement and follow an initiator should be explored in combination with how these individual differences influence the success of collective movements. While there has been significant work in this area already (Sumpter, 2009; Sueur and Deneubourg, 2011), of primary interest are motivations that either have analogues, or that can give inspiration for analogues, in multi-robot systems. Second, although the extended model accounted for local communication in following, it does not do so for canceling. Although the canceling rate is minimal for movements comprising 10 or more individuals, it is still non-zero and will require modification for it to be entirely consistent with the use of local communication. Lastly, the simulations should be extended to include actual movement. In some animal species, edge individuals exhibit greater leadership success because of the freedom of movement afforded by being on the edge of the group (Ramseyer et al., 2009). It would be interesting to learn how this freedom of movement alters the effects of local communication and topology that are predicted by the stationary simulations used here.

Acknowledgments

The author wishes to thank Ingo Schlupp, Jacques Gautrais, Blake Jordan, Elizabeth Valle, and Mo Niazi for their contributions and insights. This work was supported by NSF grant No. BCS-1124837.

References

- Ampatzis, C., Tuci, E., Trianni, V., and Dorigo, M. (2008). Evolution of signaling in a multi-robot system: Categorization and communication. *Adaptive Behavior*, 16(1):5–26.
- Anderson, M. and Papanikolopoulos, N. (2008). Implicit cooperation strategies for multi-robot search of unknown areas. *Journal of Intelligent and Robotic Systems*, 53(4):381–397.
- Balch, T. and Arkin, R. C. (1994). Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1):27–52.
- Ballerini, M., Cabibbo, N., Candelier, R., Cavagna, A., Cisbani, E., Giardina, I., Lecomte, V., Orlandi, A., Parisi, G., Procaccini, A., Viale, M., and Zdravkovic, V. (2008). Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the National Academy of Sciences*, 105(4):1232–1237.
- Bode, N., Franks, D., and Wood, A. (2012). Leading from the front? social networks in navigating groups. *Behavioral Ecology and Sociobiology*, pages 1–9.
- Couzin, I. D. (2009). Collective cognition in animal groups. *Trends in Cognitive Sciences*, 13(1):36–43.
- Drapier, M., Chauvin, C., and Thierry, B. (2002). Tonkean macaques (*Macaca tonkeana*) find food sources from cues conveyed by group-mates. *Animal Cognition*, 5(3):159–165.
- Fragaszy, D., Visalberghi, E., and Fedigan, L. (2004). *The complete capuchin: The biology of the genus Cebus*. Cambridge University Press.
- Gautrais, J. (2010). The hidden variables of leadership. *Behavioural Processes*, 84(3):664–667.
- Jacobs, A., Sueur, C., Deneubourg, J., and Petit, O. (2011). Social network influences decision making during collective movements in brown lemurs (*Eulemur fulvus fulvus*). *International Journal of Primatology*, 32:721–736.
- Kasper, C. and Voelkl, B. (2009). A social network analysis of primate groups. *Primates*, 50(4):343–356.
- Klavins, E. (2003). Communication complexity of multi-robot systems. *Algorithmic Foundations of Robotics V*, pages 275–292.
- Leca, J., Gunst, N., Thierry, B., and Petit, O. (2003). Distributed leadership in semifree-ranging white-faced capuchin monkeys. *Animal Behaviour*, 66(6):1045–1052.
- Petit, O., Gautrais, J., Leca, J.-B., Theraulaz, G., and Deneubourg, J.-L. (2009). Collective decision-making in white-faced capuchin monkeys. *Proceedings of the Royal Society B: Biological Sciences*, 276(1672):3495–3503.
- Pillot, M.-H., Gautrais, J., Arrufat, P., Couzin, I. D., Bon, R., and Deneubourg, J.-L. (2011). Scalable rules for coherent group motion in a gregarious vertebrate. *PLoS ONE*, 6(1):e14487.
- Pyritz, L., King, A., Sueur, C., and Fichtel, C. (2011). Reaching a consensus: Terminology and concepts used in coordination and decision-making research. *International Journal of Primatology*, 32(6):1268–1278.
- Ramseyer, A., Boissy, A., Dumont, B., and Thierry, B. (2009). Decision making in group departures of sheep is a continuous process. *Animal Behaviour*, 78(1):71–78.
- Sueur, C. and Deneubourg, J. (2011). Self-organization in primates: Understanding the rules underlying collective movements. *International Journal of Primatology*, 32(6):1413–1432.
- Sueur, C., Deneubourg, J., and Petit, O. (2011). From the first intention movement to the last joiner: Macaques combine mimetic rules to optimize their collective decisions. *Proceedings of the Royal Society B: Biological Sciences*, 278(1712):1697–1704.
- Sueur, C. and Petit, O. (2008). Organization of group members at departure is driven by social structure in macaca. *International Journal of Primatology*, 29(4):1085–1098.
- Sumpter, D. J. (2009). Group behaviour: Leadership by those in need. *Current Biology*, 19(8):R325–R327.
- Wey, T., Blumstein, D. T., Shen, W., and Jordán, F. (2008). Social network analysis of animal behaviour: A promising tool for the study of sociality. *Animal Behaviour*, 75(2):333–344.

Evolving a Follower in the Presence of a Potential Leader

Brent E. Eskridge

Department of Computer Science and Network Engineering, Southern Nazarene University, Bethany, OK 73008
beskridge@snu.edu

Abstract

In many real-world tasks, the ability to use a group of autonomous agents provides significant benefits over a single agent. However, these benefits come at the cost of greater complexity, particularly in the areas of cooperation and coordination. While many approaches address this problem, of particular interest is the use of leaders that emerge through action, and not group deliberation. Other agents follow these “emergent leaders” through the use of environmental cues, rather than explicit communication. While there have been many observations of emergent leadership both in natural and artificial systems, there is a lack of understanding into how this behavior can be reproduced and fostered in artificial systems. In the work presented here, experiments inspired by studies of natural systems were performed to evaluate the ease with which following behaviors could be evolved. Agent controllers were evolved both in isolation and in the presence of a potential leader. Results show that the controllers evolved in the presence of a potential leader exhibited following behaviors when there was an evolutionary advantage and did not incur a fitness penalty when doing so. In fact, agents that followed a leader agent were able to achieve higher fitness than agents acting alone in comparable situations.

Introduction

The ability to use groups of autonomous agents, or **multi-agent systems (MASs)**, in interesting, real-world tasks such as exploration, reconnaissance, and search and rescue, depends on effective coordination in complex, dynamic environments. Striking a balance between the effort required to coordinate the group and the effort required to accomplish the task is a significant problem when groups of agents are scaled beyond a few individual agents (Rosenfeld et al., 2008). Despite this problem, the use of groups provides significant performance and adaptive advantages, whether it is the improved protection from predation in natural systems or the increased fault tolerance in artificial systems.

Since coordination of large groups is frequently observed in nature, researchers are increasingly taking inspiration from the mechanisms observed in and models describing collective movement in natural systems. One mechanism that is particularly interesting is that of leadership. While leadership is not a novel idea in MASs, most leaders in

MASs are chosen *a priori* and frequently act as managers. However, leaders in natural systems often emerge from within the group based on the current situation. This is especially the case in fission-fusion societies where group membership changes frequently and long-term relationships are rare. As a result, group leadership in these systems, and the group as a whole, are able to adapt to dynamic environments and complex tasks much easier than current MASs. While most research investigating leader-follower relationships has historically focused on the aspects of leadership, a deeper understanding of followership would result in more significant progress in understanding the leader-follower relationship and in its application in MASs.

In the work described here, the ability to evolve a following behavior in the presence of both effective and ineffective potential leaders was evaluated. It was hypothesized that an agent controller which exhibited a following behavior could be evolved if it provided a clear fitness benefit. Two experiments, inspired by observations of natural systems, were performed to evaluate this hypothesis. First, a single, evolved agent was tasked with reaching maturity while avoiding a predator. The maturation task was complicated by the addition of varying levels of noise in sensing a predator. The first task acted as a baseline of comparison for the second experiment in which an evolved agent was again tasked with reaching maturity while avoiding a predator, but, in this case, had the opportunity to use the actions of a potential leader as an additional indicator of the presence of a predator. The results of these experiments confirm the hypothesis as agent controllers exhibiting following behavior were successfully evolved in relatively few generations. Furthermore, when compared to the results from the single-agent experiment, individuals that followed did not incur a fitness penalty by following a leader and, in some cases, achieved higher fitness by following.

Background

While a number of mechanisms can be used to facilitate coordination of a MAS, of particular interest is the concept of leadership. In both natural and artificial systems, the use of

a leader is frequently found to improve the cooperation and coordination of a group (Couzin, 2009; Yu et al., 2010).

Leadership

While a variety of approaches to leadership exist, this work defines a leader as an agent that initiates actions and communicates motivation implicitly through the actions it takes. In this context, the term “initiator” might be more appropriate than “leader” (Petit and Bon, 2010; Conradt and Roper, 2005), but the term “leader” is consistently used throughout the literature. Traditionally, leaders in MASs are frequently more “managers” that direct other agents, rather than the “initiator” model used here (Farinelli et al., 2004). Not only does this managerial model of leadership usually require explicit communication, but it also often makes *a priori* assumptions about the distribution of knowledge and capabilities within the MAS. These assumptions present problems in the dynamic environments and complex tasks in which MASs can provide the most benefit. In contrast, in the initiator model of leadership, leaders are not chosen, rather, they emerge from within the group by virtue of the fact that others choose to follow them. There are a variety of reasons for a particular agent emerging as a leader, including a behavioral trait, a morphological trait, or unique access to information that increases the emergent leader’s motivation to act first (King et al., 2009). As a result, other agents observe the actions of the emergent leader and determine that it is in their best interests to follow. Emergent leadership is observed frequently in natural systems, including fish (Harcourt et al., 2009), sheep (Pillot et al., 2009), ravens (Marzluff et al., 1996), and crows (Sonerud et al., 2001). It has even been shown to emerge in MASs (Nouyan et al., 2009; Ghijsen et al., 2010), but those studies did not focus their investigation on the emergent aspect of leadership.

Communication

Traditional models of leadership in MASs usually rely on significant communication, which is frequently categorized as either being explicit or implicit. Explicit communication can be defined as the intentional signaling of information through a defined protocol. Implicit communication, on the other hand, can be defined as an indirect method of communication that uses the individual’s actions, and the resulting changes in the environment, to communicate information. Environmental cues, frequently observed in natural systems, are an example of implicit communication, in which an individual learns “to associate a behavior, a trace, or an object with the occurrence of a given event” (Drapier et al., 2002). Using implicit communication, an informed individual communicates information through its actions, or how its actions modify the environment (Sumpter, 2010).

While explicit communication is commonly used and can be an effective method of improving cooperation and coordination in a MAS (Balch and Arkin, 1994; Ampatzis et al.,

2008), there can be significant problems with its use. Not only is explicit communication less flexible and more sensitive to environmental conditions than implicit communication, it has problems scaling to large numbers of agents (Anderson and Papanikolopoulos, 2008). Dependence on explicit communication is a significant point of failure and leaves the system vulnerable to a new set of problems such as interference and message authentication. Furthermore, there are many situations where explicit communication in a MAS is physically not possible, not practical, too expensive, too complex to be effective in large teams, or may compromise the system’s ability to accomplish a task. Since explicit communication presents significant problems and is not necessary when implicit communication is available (Balch and Arkin, 1994), implicit communication is increasingly being used instead (Pereira et al., 2002; Ampatzis et al., 2008; de Greeff and Nolfi, 2010). While implicit communication has the potential for lower performance than explicit communication, it is a far more practical choice for larger MASs. Implicit communication is simpler, is more robust to change, has lower power consumption, and is stealthier than explicit communication (Pereira et al., 2002; Anderson and Papanikolopoulos, 2008).

Motivation

There are two components to the development of a follower behavior that warrant investigation. The first is the development of the follower behavior itself, and was the focus of the work presented here. The second is the development of the decision-making process that results in choosing to use the follower behavior, and will be the subject of future work. In an effort to evaluate the development of a follower behavior, the simulations described here use a highly abstract evaluation environment to model a maturation problem inspired by a number of experiments in and observations of natural systems. The intent of using an abstract environment is to minimize the number of confounding variables, while retaining the essential aspects of the systems observed in nature.

Simple Maturation Experiment

For this experiment, a simulated agent was tasked with maturing into adulthood, while avoiding predators. In this first experiment, a single agent acted alone, without the benefit of a potential leader. As such, this experiment provided a baseline against which the results of the subsequent MAS experiment with a potential leader could be compared. It was modeled after studies of activity levels of larval anurans (i.e., tadpoles), salamanders, and fish in the presence of predators (Richardson, 2001; Sih et al., 2003; Harcourt et al., 2009) and game-theoretic models of leadership-followership decisions (Rands et al., 2003). As in the natural systems, the agent in this experiment had to risk capture by predators to forage for food, which both ensured its continued survival and enabled the maturation process. Once the agent reached

Parameter	Value
$e_{consumed}$	0.04
e_{exist}	0.01
e_{gain}	0.01
$e_{initial}$	0.50
e_{max}	1.00
m_{energy}	0.02
$m_{threshold}$	0.60
p_{period}	80
Max timesteps	500
Population	100
Generations	100
Mutation rate	1%

Table 1: Experimental parameters used for the maturation experiment are shown.

full maturity, it was no longer considered to be at risk of capture by the predators, which is consistent with natural systems where mature individuals are generally not vulnerable to the same predators as they were during maturation.

Experimental Setup

For this simulation, the act of foraging for food was abstracted into a single value representing an agent's activity level. At each timestep, the agent gained energy based on its activity level, calculated by the following

$$E_{gain} = a \cdot e_{gain} \quad (1)$$

where a was the agent's activity level lying in the range $[0, 1]$ and e_{gain} was the maximum amount of energy that could be gained by foraging (see Table 1). When the agent had a high activity level, it was considered to be foraging and gained energy. When the agent had zero activity, it was considered to be at rest. The energy gained was then added to the agent's energy reserves, referred to as the energy level and denoted E_{total} with a range of $[0, 1]$. Also, at each timestep, the agent consumed energy as a result of its activity level and the energy costs associated with living. The amount of energy consumed by an agent at each timestep was calculated by the following

$$E_{consumed} = a \cdot e_{consumed} + e_{exist} \quad (2)$$

where a was, again, the agent's activity level, $e_{consumed}$ was the maximum amount of energy consumed by foraging, and e_{exist} was the energy costs for the agent's existence. The energy consumed was then subtracted from the agent's energy level. If the agent's energy level ever dropped below zero, it was considered to have died and the trial was terminated.

If the agent's energy level exceeded a threshold value, specified by $m_{threshold}$, a portion of the energy was used

to mature the agent. The amount of energy used for maturation was calculated by the following

$$E_{maturation} = \min\{m_{energy}, E_{total} - m_{threshold}\} \quad (3)$$

where m_{energy} was the default amount of energy used, E_{total} was the current energy level of the agent, and $m_{threshold}$ was, again, the threshold value for maturation. This ensured that the agent only used the energy exceeding the threshold for maturation and did not use energy that was reserved for maintenance (i.e., foraging and existence). This is consistent with observations of energy allocation in natural systems (Heino and Kaitala, 1999). The maturation energy was transferred from the agent's energy level to its maturation level, denoted M with a range of $[0, 1]$. When the agent's maturation level met or exceeded 1.0, it was considered to have fully matured and the trial was terminated.

Predation was modeled as a single value, denoted p_{level} , that indicated the current level of predation and cycled between periods of high and low predation with a value in the range $[0, 1]$. This was considered to be a general indication of the activity level of predators in the vicinity of the agent, and did not represent a specific predator. The predation level at timestep t was calculated by the following

$$p_{level} = \left(\frac{\sin\left(\frac{2\pi t}{p_{period}}\right) + 1}{2} \right)^2 \quad (4)$$

where p_{period} was the period of the predation cycle. The predation value was squared to ensure that there were enough opportunities to forage with minimal predation, while still retaining times of high predation. The agent's probability of being captured by a predator at a given timestep was calculated by the following

$$P_{capture} = a \cdot p_{level} \quad (5)$$

where a , again, was the agent's activity level and p_{level} was the predation level. Thus, an agent was free to forage and have a high activity level if the predation level was low, while it was risky to forage when the predation level was high. At each timestep, a random number drawn from the uniform distribution in the range $[0, 1]$ was generated to determine if the agent had been captured. If the random number exceeded $P_{capture}$, the agent was classified as having been captured by a predator and the trial was terminated.

To introduce uncertainty into the simulation, the agent's ability to sense the current predation level was restricted by the addition of sensor "noise." Given the abstract nature of this experiment, this simplified sensing model provided the ability to tune the agent's uncertainty in its knowledge of the environment without introducing too many confounding variables. Sensor noise was modeled as an offset applied to the actual predation level, p_{level} , and was a random number, drawn from a Gaussian distribution with mean 0 and

standard deviation of 1, denoted $N(0,1)$. In each treatment, the randomly generated offset was multiplied by a value less than or equal to 1, denoted p_{noise} , which served to reduce the standard deviation of the Gaussian distribution from which the offset was drawn. The sensed predation level at each timestep was calculated by the following

$$p_{sensed} = p_{level} + p_{noise} \cdot N(0,1) \quad (6)$$

where p_{sensed} was the sensed predation level, p_{level} was, again, the actual predation level, p_{noise} was the sensor noise level, and $N(0,1)$ is a random value drawn from the previously described Gaussian distribution. Hereafter, this noise value is referred to as a sensor noise percentage (i.e., “X%”), with $p_{noise} = 1$ being referred to as “100%.” In some treatments, the agent’s predation sensor was modeled as having completely failed. In these situations, which are denoted by a sensor noise level of “Random,” the sensor produced random values drawn from a uniform distribution in the range $[0,1]$, denoted $U(0,1)$. This represented a worst-case scenario in which the sensed predation level was completely unpredictable, unlike the other treatments in which random noise was added to a known good predation level.

An agent’s decision-making was performed by an **artificial neural network** (ANN) that was evolved using FS-NEAT (Whiteson et al., 2005), a variation on the standard NEAT algorithm (Stanley and Miikkulainen, 2002) in which ANNs in the initial population have no hidden nodes and no connections between nodes except those added by an initial mutation. The inputs to the ANN were a bias signal, the agent’s energy level, the agent’s maturation level, and the current predation level. The output was the agent’s activity level and was also normalized to the range $[0,1]$. The weights of the evolved ANNs were fixed once created.

Treatments with sensor noise percentages ranging from 0% to 50% were evaluated. Two additional treatments were used that represented worst case scenarios for the agent. In the first, the sensed predation level was completely random. In the second, the predation level sensor was missing and the ANN received a constant input of 0 for each timestep. Forty experimental runs were performed for each treatment with ANNs evaluated in five trials. Fitness was calculated as the mean final maturation level of the agent in each trial when the trial ended. While the time required for an agent to mature was not a part of the fitness calculation, there was an implicit benefit for faster maturation since it resulted in lower maintenance costs and afforded fewer opportunities for a predator to capture the agent. Furthermore, a capture did not preclude an ANN from being selected as a parent for the next generation. While this would be the case in a natural system, it is important to remember that it was the evolved ANN that was evaluated and received fitness, not the agent itself. Table 1 shows the experiment-specific parameter settings that were used. NEAT-specific parameter settings were based on standard NEAT defaults and are re-

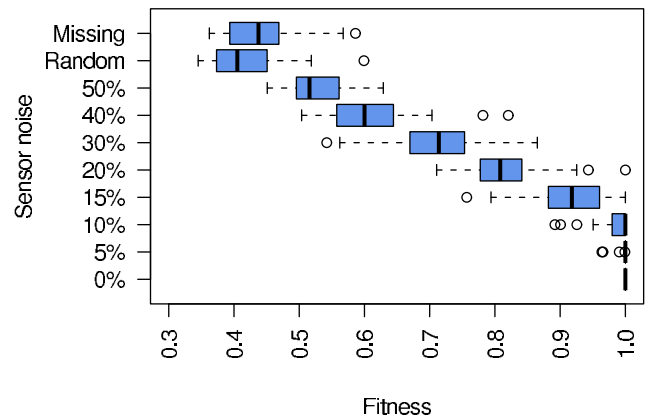


Figure 1: The mean best-of-run fitness values over all five trials for varying levels of sensor noise in the single agent maturation experiment treatments are shown. For each treatment, the box depicts the interquartile range (IQR) from the first quartile to the third quartile over all 40 trials, the vertical line represents the median fitness value, the whiskers represent the ± 1.5 IQR, and the circles represent outlier values.

ported elsewhere (Stanley and Miikkulainen, 2002).

Results

Figure 1 shows the mean best-of-run fitness values over five trials for the single-agent experiment using boxplots. For each treatment, the box depicts the interquartile range (IQR) from the first quartile to the third quartile over all 40 trials, the vertical line represents the median fitness value, the whiskers represent the ± 1.5 IQR, and the circles represent outlier values (Robbins, 2005). All experimental runs were able to evolve an ANN that resulted in the successful maturation of an agent in all five trials when no sensor noise was added to the agent’s predation level precept (i.e., 0% sensor noise). The performance of the ANNs dropped dramatically as sensor noise was added. However, consistently successful ANNs were evolved even with the addition of small amounts of sensor noise. Results of the bootstrapped Kolmogorov-Smirnov¹ test show that there was no statistically significant difference in the fitness values between the 0% and 5% sensor noise treatments ($p = 0.081$), while there was a statistically significant difference between the 0% and the remaining sensor noise treatments ($p \approx 0$).

Leader Experiment

One reason for the emergence of leaders in natural systems is an individual possessing knowledge that other individuals in the group do not (King et al., 2009). Since incorrect information, in the form of a “noisy” predation sensor, was

¹Unlike other standard significance tests, the KS test does not assume that the data has a normal distribution. Since these results suffer from ceiling effects, this assumption cannot be made.

already present in the simulation, potential leaders were introduced as potentially having more accurate information on the current level of predation.

As in the first experiment, an evolved agent was tasked with surviving until maturity. However, in this experiment, a potential leader, whose actions the evolved agent could observe, was added to the environment. The evolved agent was, therefore, able to observe the activity of the potential leader in response to the current predation level. As a result, the evolved agent could exhibit a following strategy if the leader's actions provided a better indicator of the current predation level than its own sensor. This provided an opportunity to determine the ease with which effective followers could be produced in response to potential leaders.

Experimental Setup

To enable the evolved agent to sense the potential leader's activity level, the potential leader's activity level was added as an input to the ANN configuration used in the single-agent experiment. Note that the use of an input for the activity level of the potential leader represents an observation by the agent of the potential leader, and not explicit communication between the potential leader and the agent. The leader's actions were controlled by a randomly chosen, best-of-run ANN from the single-agent experiment with no sensor noise. As a result, the potential leader was unable to sense the activity level of the agent under evaluation.

In each treatment, Gaussian noise was added to the agent's predation level sensor, the potential leader's predation level sensor, or both. The addition of sensor noise to the potential leader's sensor was used to evaluate the evolved agent's performance in the absence of a perfectly accurate leader. To ensure that the evolved agent always received some environmental cues from the potential leader, the potential leader was ineligible for capture by the predator, regardless of the quality of its actions. While this may not be a realistic or viable long-term assumption, it proved to be an effective simplification given the highly abstract nature of the current experiment and will be revisited in future experiments that use more realistic, high-fidelity environments. The experiment-specific parameter settings used in this experiment were the same as the single-agent experiment and are shown in Table 1.

Results

Figure 2 shows the mean best-of-run fitness values for experimental treatments in which a potential leader was present in the environment. In treatments where at least one of the two agents, either the agent under evaluation or the potential leader, had 0% sensor noise, evolved ANNs were consistently able to produce behaviors that resulted in agents successfully maturing in each of the five separate trials. Agents were captured by a predator in only a few trials (see Figure 2a). An analysis using the bootstrapped Kolmogorov-

Smirnov test shows that there was no statistically significant difference between any of the treatments, including treatments with a few outliers in which the agent was captured before reaching full maturation. This indicates that, when appropriate, the evolved ANN was able to use either its own predation level percepts or the activity level of the potential leader with equal effectiveness. One particular treatment of note is the one in which the evolved ANN's predation level sensory input was completely missing. Since the ANNs evolved in this treatment did not differ in fitness from the single-agent treatment, it can be concluded that using the potential leader's activity level as a proxy for the predation level did not incur any inherent fitness penalty.

The results for treatments in which the minimum sensor noise level were 5% are consistent with the previous set in that evolved ANNs were able to use either a direct sensing of the predation level or the potential leader's activity level as a predation level indicator with equal effectiveness (see Figure 2b). However, in this case, evolutionary runs in three treatments achieved higher fitness than similar treatments in the single-agent experiment. In evolutionary runs in which both agents had 5% sensor noise or one agent had 5% sensor noise and the other had 10% sensor noise, evolved ANNs were able to produce behavior that resulted in the successful maturation of an agent in each of the five trials over all forty experimental runs. Although the significance level between fitness values was relatively high ($p < 0.1$) and the difference in fitness was relatively small (0.998 ± 0.008 vs. 1.0 ± 0.0), it bears mentioning since it indicates a trend that will be observed in later treatments.

In the results for treatments in which the minimum sensor noise level was 10%, similar results were found (see Figure 2c). For only the treatment in which the evolved agent had 10% sensor noise and the potential leader's predation level input was random did the addition of the potential leader result in statistically significantly *lower* fitness ($p = 0.001$). Although the relative differences in fitness were slight, the reason for this drop in fitness is unknown and warrants further investigation. Similar to the previous set of treatments, the treatment in which both agents had 10% sensor noise had statistically significantly higher fitness than the single-agent treatment with 10% noise ($p = 0.002$).

Lastly, the results for treatments in which the minimum sensor noise level was 15% were consistent with previous treatments (see Figure 2d). The treatments in which one agent had 15% sensor noise and the other had a random input for the predation level produced results comparable to the single-agent treatment and were not statistically significantly different. The treatment in which both agents had 15% sensor noise had statistically significantly higher fitness, as in the previous treatment sets ($p < 0.0001$).

To further investigate the phenomenon in which agents in the leadership experiment were able to achieve higher fitness than a single agent with the same level of noise, the rate at

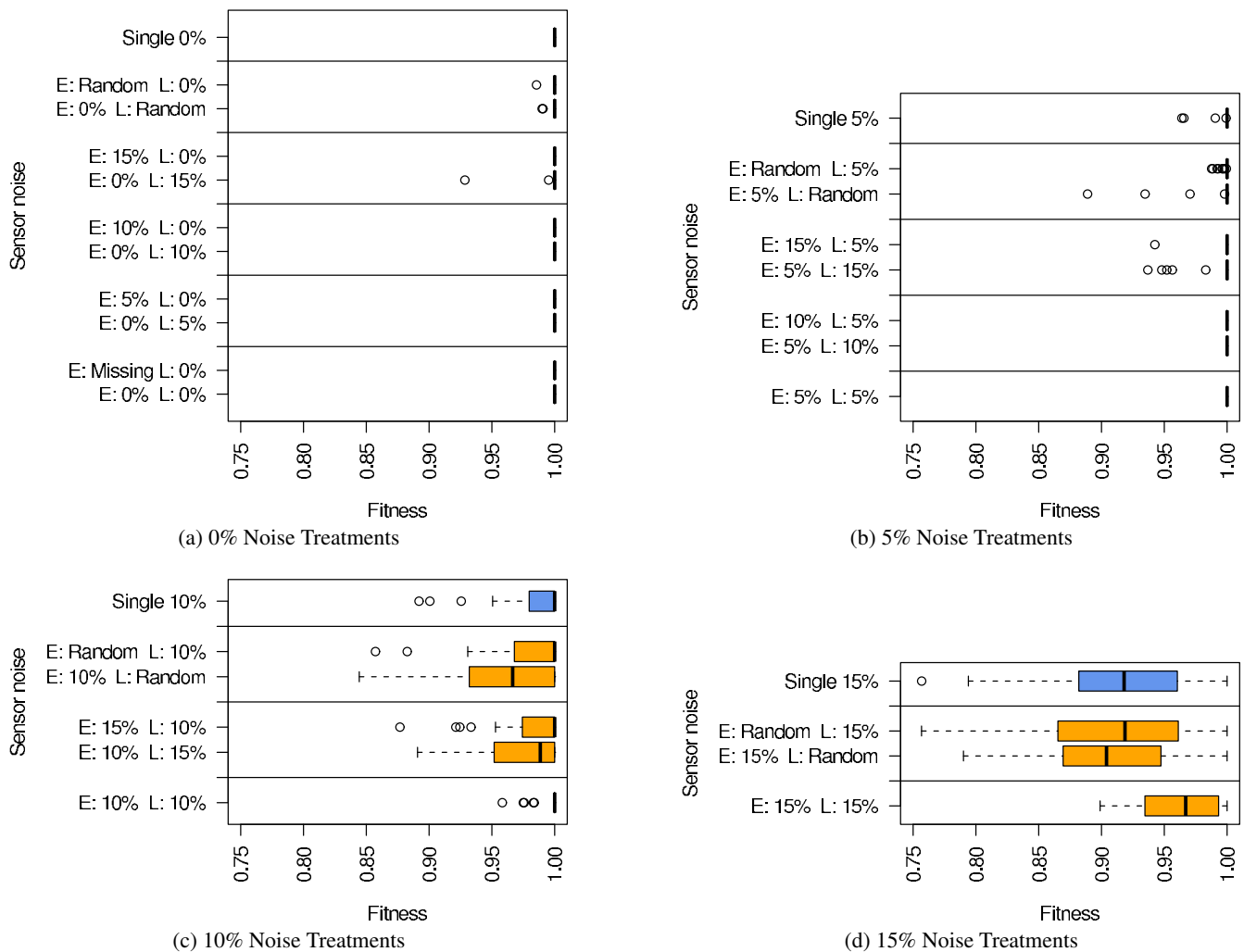


Figure 2: The mean best-of-run fitness over five trials and forty runs for the maturation experiment in the presence of a potential leader are shown. Individual treatments are organized by sensor noise levels and are compared to results from the single-agent experiment with similar sensor noise levels. The “E” term represents the sensor noise of the evolved agent and the “L” term represents the sensor noise level of the potential leader.

which fit ANNs evolved in each treatment were compared using the randomized two-way ANOVA test (Piater et al., 1998). In each case, evolved ANNs from the leadership experiment treatment achieved higher fitness faster than the treatment from the single-agent experiment with $p = 0.05$. Figure 3 illustrates these results in a comparison of the fitness curves between treatments.

Discussion

There are two main conclusions that can be drawn from these results. First, ANNs were evolved in a few generations that were capable of using the observed leader’s actions as an indicator of the current predation level if their own sensor percepts were unreliable. Although increased noise in the predation sensor made the evolution of follow-

ing behaviors more difficult, the fitness curves in Figure 3 show that effective controllers exhibiting following behaviors were evolved in under 100 generations for up to 15% sensor noise. In treatments for which following was a viable strategy, the mean generation at which the best-of-run ANN was found was 34.9 with a standard deviation of 16.5. Furthermore, this following behavior did not incur a fitness penalty as some of the treatments in the 0% sensor noise treatments shown in Figure 3a illustrate.

Second, these results demonstrate that an effective following behavior can result in performance that is superior to performance in the single-agent experiment. This superior performance is shown not only in the statistically significantly higher fitness of the evolved ANNs, but also in the statistically significantly faster rate in which fit ANNs were

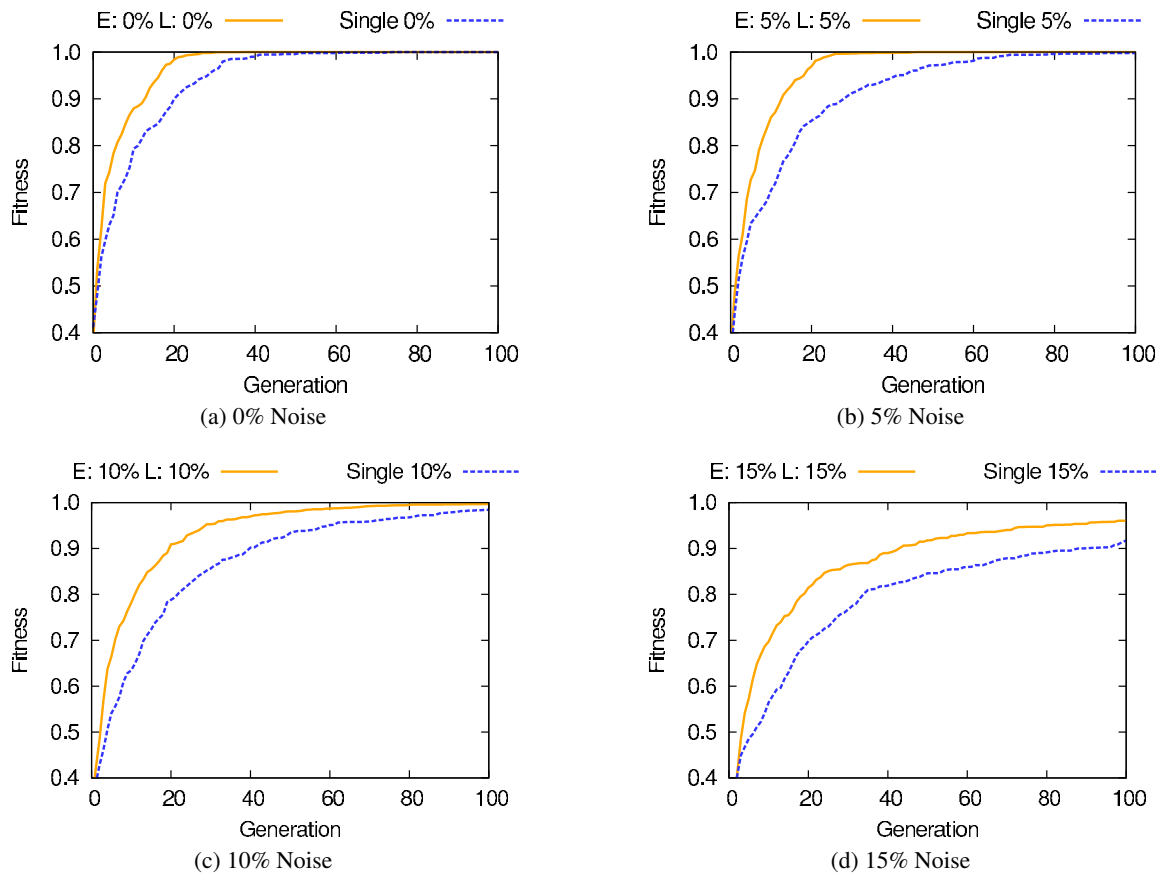


Figure 3: Plots of the mean best-of-run fitness at each generation over all forty experimental runs for selected treatments of the single-agent and leadership experiments are shown. Selected treatments compare the fitness of treatments in the single-agent experiment with treatments in the leadership experiment in which both agents had the same sensor noise level.

evolved. While this phenomenon of higher group performance, referred to as the “many wrongs principle,” is observed in group navigation found in nature (Simons, 2004), its observation was not expected in this highly abstract environment, as there were only two agents present and the potential leader was ignorant of the evolved agent’s presence.

Conclusions

For MASs to be useful in the dynamic, real-world tasks for which they can provide the most benefit, the problem of effectively coordinating even moderate numbers of agents must be solved. One promising approach is through the use of “emergent leadership.” In emergent leadership, leaders arise from within a group by virtue of the actions that they take, and do not require extensive communication. For emergent leadership to work, however, other agents within the group must decide to follow the leader. In the work presented here, the ability to evolve agent controllers capable of exhibiting following behaviors was evaluated. The experimental results demonstrate that the effective controllers were evolved in relatively few generations, even in the pres-

ence of sensor noise, and without the benefit of explicit communication between the leader and follower. Furthermore, following did not incur a loss in performance when compared to the single-agent simulations and even resulted in a performance increase in some simulations.

There are a variety of directions for future work. First, the work presented here used a static environmental configuration in which it was either beneficial or not beneficial to follow the leader. As noted above, the second component of interest in followership is the development of an effective decision-making process. Additional experiments indicate that attempting to evolve both a following behavior and the decision-making involved in deciding to follow at once can be too complex to evolve in a single ANN. Further work is necessary to ensure that agents can adapt to dynamic environments in which the benefits to following vary with time and the decision to follow is less clear-cut. Also, in these experiments, the agent was presented with a simple choice: follow a single known agent or follow no one. When an agent is a member of a much larger group, it is presented with the much more difficult choice of deciding *which* agent

to follow, if one at all. Further work is also required to determine if the relative ease with which following behaviors were evolved persists as the number of agents is scaled up.

Acknowledgments

The author wishes to thank Ingo Schlupp and Dean Hougen for their contributions and insights. This work was supported by NSF grant No. BCS-1124837. Some of the computing for this project was performed at the OU Supercomputing Center for Education & Research (OSCAR) at the University of Oklahoma.

References

- Ampatzis, C., Tuci, E., Trianni, V., and Dorigo, M. (2008). Evolution of signaling in a multi-robot system: Categorization and communication. *Adaptive Behavior*, 16(1):5–26.
- Anderson, M. and Papanikolopoulos, N. (2008). Implicit cooperation strategies for multi-robot search of unknown areas. *Journal of Intelligent and Robotic Systems*, 53(4):381–397.
- Balch, T. and Arkin, R. C. (1994). Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1):27–52.
- Conradt, L. and Roper, T. (2005). Consensus decision making in animals. *Trends in Ecology & Evolution*, 20(8):449–456.
- Couzin, I. D. (2009). Collective cognition in animal groups. *Trends in Cognitive Sciences*, 13(1):36–43.
- de Greeff, J. and Nolfi, S. (2010). Evolution of implicit and explicit communication in mobile robots. In *Evolution of Communication and Language in Embodied Agents*, pages 179–214. Springer.
- Drapier, M., Chauvin, C., and Thierry, B. (2002). Tonkean macaques (*Macaca tonkeana*) find food sources from cues conveyed by group-mates. *Animal Cognition*, 5(3):159–165.
- Farinelli, A., Iocchi, L., and Nardi, D. (2004). Multirobot systems: a classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(5):2015–2028.
- Ghijsen, M., Jansweijer, W. N. H., and Wielinga, B. J. (2010). *Adaptive Hierarchical Multi-agent Organizations*, volume 281 of *Studies in Computational Intelligence*, pages 375–400. Springer.
- Harcourt, J. L., Ang, T. Z., Sweetman, G., Johnstone, R. A., and Manica, A. (2009). Social feedback and the emergence of leaders and followers. *Current Biology*, 19(3):248–252.
- Heino, M. and Kaitala, V. (1999). Evolution of resource allocation between growth and reproduction in animals with indeterminate growth. *Journal of Evolutionary Biology*, 12(3):423–429.
- King, A. J., Johnson, D. D., and Van Vugt, M. (2009). The origins and evolution of leadership. *Current Biology*, 19(19):R911–R916.
- Marzluff, J. M., Heinrich, B., and Marzluff, C. S. (1996). Raven roosts are mobile information centres. *Animal Behaviour*, 51(1):89–103.
- Nouyan, S., Groß, R., Bonani, M., Mondada, F., and Dorigo, M. (2009). Teamwork in self-organized robot colonies. *IEEE Transactions on Evolutionary Computation*, 13(4):695–711.
- Pereira, G., Pimentel, B., Chaimowicz, L., and Campos, M. (2002). Coordination of multiple mobile robots in an object carrying task using implicit communication. In *International Conference on Robotics and Automation*, pages 281–286. IEEE.
- Petit, O. and Bon, R. (2010). Decision-making processes: The case of collective movements. *Behavioural Processes*, 64(3):635–647.
- Piater, J. H., Cohen, P. R., Zhang, X., and Atighetchi, M. (1998). A randomized ANOVA procedure for comparing performance curves. *International Conference on Machine Learning*, pages 430–438.
- Pillot, M. H., Gautrais, J., Gouello, J., Michelena, P., Sibbald, A., and Bon, R. (2009). Moving together: Incidental leaders and naïve followers. *Behavioural Processes*, pages 235–241.
- Rands, S., Cowlshaw, G., Pettifor, R., Rowcliffe, J., and Johnstone, R. (2003). Spontaneous emergence of leaders and followers in foraging pairs. *Nature*, 423(6938):432–434.
- Richardson, J. M. (2001). A comparative study of activity levels in larval anurans and response to the presence of different predators. *Behavioral Ecology*, 12(1):51–58.
- Robbins, N. (2005). *Creating more effective graphs*. Wiley-Interscience.
- Rosenfeld, A., Kaminka, G., Kraus, S., and Shehory, O. (2008). A study of mechanisms for improving robotic group performance. *Artificial Intelligence*, 172(6-7):633–655.
- Sih, A., Kats, L. B., and Maurer, E. F. (2003). Behavioural correlations across situations and the evolution of antipredator behaviour in a sunfish-salamander system. *Animal Behaviour*, 65(1):29–44.
- Simons, A. (2004). Many wrongs: the advantage of group navigation. *Trends in Ecology & Evolution*, 19(9):453–455.
- Sonerud, G., Smedshaug, C., and Bråthen, Ø. (2001). Ignorant hooded crows follow knowledgeable roost-mates to food: support for the information centre hypothesis. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1469):827–831.
- Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.
- Sumpter, D. J. (2010). *Collective Animal Behavior*. Princeton University Press.
- Whiteson, S., Stone, P., Stanley, K. O., Miikkulainen, R., and Kohl, N. (2005). Automatic feature selection in neuroevolution. In *Genetic and Evolutionary Computation Conference*, pages 1225–1232.
- Yu, C.-H., Werfel, J., and Nagpal, R. (2010). Collective decision-making in multi-agent systems by implicit leadership. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 1189–1196. IFAAMAS.

An ecology-based evolutionary algorithm to evolve solutions to complex problems

Sherri Goings¹, Heather Goldsby², Betty H.C. Cheng³, and Charles Ofria³

¹Computer Science Dept., Carleton College, MN, 55057

²University of Washington, Seattle, WA, 98195

³Department of Computer Science & Engineering, Michigan State University, East Lansing, MI, 48824
sgoings@carleton.edu

Abstract

Evolutionary algorithms have shown great promise in evolving novel solutions to real-world problems, but the complexity of those solutions is limited, unlike the apparently open-ended evolution that occurs in the natural world. In part, nature surmounts these complexity barriers with ecological dynamics that generate a diverse array of raw materials for evolution to build upon. The authors previously introduced *Eco-EA*, an evolutionary algorithm that integrates these natural ecological dynamics to promote and maintain diversity in the evolving population. Here, we apply the *Eco-EA* to the real-world software engineering problem of evolving behavioral models for deployed nodes in a remote sensor network for flood monitoring. We show that the *Eco-EA* evolves good behavioral models faster than a traditional EA, generates a more diverse suite of models than a traditional EA, and creates models that are themselves more evolvable than those created by a traditional EA.

Introduction

Evolutionary algorithms (EAs) have shown great promise in evolving novel solutions to real-world problems, but the complexity of those solutions is limited, unlike the apparently open-ended evolution that occurs in the natural world. In part, nature surmounts these complexity barriers with natural ecological dynamics that generate an incredibly diverse array of raw materials for the evolutionary process to build upon, the efficacy of which has been demonstrated in the artificial life system *Avida* (Cooper and Ofria, 2002).

For EAs to solve more complex problems, we must study how highly complex traits arise in the natural world, and where EAs fall short in duplicating these dynamics. The complexity of solutions produced by traditional EAs is typically limited by rapid convergence to a single solution on a sub-optimal local peak, resulting in stagnation. EA researchers recognize the importance of maintaining variation in evolving populations to prevent stagnation and make use of a variety of diversity preserving techniques. However, it has proven difficult to reach the levels of species density and variety found in nature (such as in bio-films (Tyson et al. 2004) or biodiversity hotspots like rain forests (Gaston 2000)) or even the high intra-species variance of individual values for a given trait. In nature, simple ecological forces promote this diversity, due to both spatial and temporal environmental heterogeneity, combined with negative frequency-dependent selection (Tilman, 1982).

Diversity in a population can provide other significant advantages beyond forestalling stagnation. Potential benefits to evolutionary algorithms include: (1) maintenance of a selection of good solutions for the researcher to choose from, often with slightly different properties; (2) representation across a Pareto front for multi-objective optimization problems; (3) the use of different partial-solutions as starting points to build the full solution from, without the researcher needing to know the ideal path; (4) resilient solutions that can withstand environmental changes; and (5) significantly more rapid evolution of targeted complex functions. Robust ecological communities exhibit all of these traits.

The authors previously introduced a method to integrate ecological factors promoting diversity into an EA using limited resources, and showed that populations evolved with this method were able to find and cover multiple niches in a simple string matching problem (Goings and Ofria, 2009). Here, we apply this new ecology-based evolutionary algorithm (*Eco-EA*) to a real-world problem in software engineering, and show that this approach yields several advantages over a traditional EA, including:

1. faster evolution of satisfactory solutions
2. evolution of a more diverse array of solutions
3. creation of solutions with greater evolvability that are easily adapted to succeed in different environments.

These results indicate that the ecology-based EA facilitates the evolution of solutions to complex problems.

Background

Eco-EA

As demonstrated by (Cooper and Ofria, 2002), forcing individuals to compete for multiple limited resources will force a population to maintain higher levels of diversity. A traditional EA can be thought of as having only one resource, where each individual's fitness is determined by the amount of that resource it can obtain. In most cases, the population size in an EA is fixed, thus making space its only limited resource (which organisms claim as they replicate). In the *Eco-EA* proposed by the authors in (Goings and Ofria, 2009), each function performed by an individual is associated with a distinct resource. When an individual performs a function it

receives a predetermined fraction of the currently available amount of the associated resource and its fitness is increased proportionately. These resources are set up as the computational equivalent of a well-stirred chemostat; that is, each resource flows into the environment at a constant rate, and a small percentage of the available resource flows out, limiting the total accumulation. Exploration of new areas of the fitness landscape is highly rewarded as an unused resource will accrue in quantity; as such, the individual first to discover the resource will receive a large fitness boost. However, when many organisms perform functions that consume the same resource, the availability of that resource will decrease until further organisms who attempt to draw from it do not receive enough reward to offset the opportunity cost of targeting a different resource.

Eco-EA in Avida

The experiments performed in this study used the Avida digital evolution research platform (Ofria and Wilke, 2004). Avida maintains a population of asexual self-replicating computer programs (“digital organisms”) that exist in a computational environment and are subject to mutations and natural selection. Each digital organism has a genome that is a sequence of instructions in a special-purpose programming language. As in natural organisms, this genome specifies the behavior of the individual. Typically, in Avida, this behavior includes the replication of the organism, but for this study we used an explicit fitness function and organisms were replicated in time inversely proportional to their fitness (i.e. higher fitness yields faster replication), similar to the process of a steady-state evolutionary algorithm. This change removed the extra selection pressure for organisms to improve their replication mechanism and simplified the analysis of individual organisms. Random mutations occur during replication and include substitutions, insertions, and deletions. The Avida instruction set is designed so that mutations always yield a syntactically correct program, albeit one that may not perform any meaningful computation. When an organism replicates, its offspring replaces a randomly chosen individual currently in the population. Thus Avida maintains a constant population size.

The environment used in this study contains a set of resources, each of which corresponds to a user-defined task. An organism must perform a task to receive a portion of the available corresponding resource. The fitness of an organism is determined by how much of each resource it consumes. In most Avida studies, as with most evolutionary algorithms, resources are unlimited, creating a single-niche environment where an organism receives a fixed amount of resource for each task completed. Thus, the fitness gained for completing a task is constant and does not reflect how many other organisms are also performing that task. In this study, however, we incorporate the ecological factor of limited resources to create a multi-niche environment that encourages the evolving population to diversify.

Avida-MDE

For this study, we use a software engineering extension to Avida called Avida-MDE (Avida for Model-Driven Engineering), previously developed by Goldsby and Cheng

(Goldsby and Cheng, 2008a). We briefly describe the motivation for the creation of Avida-MDE, establish its links to real-world problems, and provide a high-level overview of how it uses Avida to automate software engineering research.

Model-driven engineering is a leading software engineering approach to developing complex software-based systems, including on-board control software for automotive and flight systems, ecosystem monitoring, and robotic systems. Many of these systems are considered high-assurance, meaning that they must satisfy safety requirements under a variety of environmental conditions. Model-driven engineering works by systematically refining graphical models that can be analyzed for adherence to requirements using a variety of analysis tools, and then automatically used to generate code (Schmidt, 2006). Konrad *et al.* have proposed a modeling and analysis process for such high-assurance systems (Konrad *et al.* 2007) where a system is represented by a class diagram that captures the structural elements and several behavioral models. A given behavioral model comprises a set of state diagrams, one for each class in the class diagram, and represents the behavior of the system under specific environmental conditions.

Manually developing the behavioral models for a system can be tedious and error prone, since each model must be created independently and it requires the developer to have foreknowledge of the possible environmental conditions. Avida-MDE is a digital evolution tool that automates this process by generating a suite of behavioral models given information from the class diagram (Goldsby and Cheng, 2008b). At a high level, Avida-MDE accepts a list of triggers, guards, and actions (created using class diagram elements) as input. These inputs are provided to each digital organism, which uses them as raw material for constructing a set of state diagrams. A new genetic language was implemented in Avida-MDE to enable organisms to manipulate the state diagrams and thus change the behavior of the model it generates. The details of this language and how the digital organisms generate models can be found in (Goldsby and Cheng, 2008b). The key concept is that a mutation to an organism’s genome changes the behavioral model that it creates.

To evaluate the generated behavioral models (and thus the organisms themselves), Avida-MDE uses a suite of software engineering tools. Several tasks were added to the Avida environment, which have previously been linked only to unlimited resources. *Software engineering metric tasks*, such as minimizing the number of transitions and maximizing the number of deterministic states, guide the evolutionary process to generate models that adhere to commonly advocated software engineering practices. *Scenario tasks* reward organisms for creating models that support one desired execution path, or scenario. Scenarios encapsulate small excerpts of model behavior that can be combined and expanded to achieve the desired overall system behavior. To account for the uncertainty in the execution environment, a developer can specify two types of scenarios; (1) *required functional scenarios* must be supported by the generated models; (2) *non-functional (NF) scenarios* each of which specify a different way to achieve the same functional objective with different non-functional characteristics (e.g., quality, reliability). A model must support at least one of each

type of NF scenarios. The specific NF scenario supported by a model impacts its non-functional behavior. Next, *witness property tasks* reward models for having at least one execution path that supports a desired system property. Lastly, *property tasks* are included to reward models for having all possible execution paths support a desired system property. For example, “no data is ever lost,” “battery levels never drop below a threshold value,” or “water level never exceeds a maximum value.”

Grid-Stix

Avida-MDE was previously used to generate behavioral models for Grid-Stix, a light-weight flood warning system that comprises a set of sensor nodes. Grid-Stix is used to monitor the water levels for potential flood conditions with the River Ribble in England (Hughes et al. 2006). Flooding is an increasing and costly problem for the United Kingdom, and early flooding predictions enable fast responses to avert flood damage. However, prediction accuracy must be balanced by two other non-functional considerations: energy efficiency (because sensor nodes have a limited power supply) and fault-tolerance (because sensor nodes are deployed remotely). The objective of the case study was to generate a suite of behavioral models for a single sensor node, where the models make different non-functional tradeoffs (i.e., different combinations of energy efficiency, prediction accuracy, and fault-tolerance) and yet all satisfy the overall functional objective of monitoring the river to collect data and pass it along to nearby nodes.

Different scenario tasks captured different non-functional tradeoffs. Specifically, three tasks rewarded models that supported scenarios for setting different processor speeds while completing various functions on the sensor, and six tasks rewarded models that supported scenarios where the sensor used different data transmission methods. A model needs to only have one path that performs a scenario behavior in order to receive the associated reward, and can receive a partial reward for partial completion of a scenario. For example, one scenario required a node to set its processor speed to 100, then query the pressure sensor at this speed for the water depth, and finally to set its depth data to the query result. A model received 50% of this scenario task reward if it set its processor speed to 100, 75% if it also queried the pressure sensor, and 100% if it completed the entire scenario.

Witness and property tasks built upon the scenario tasks to reward for desired overall system behavior; for example sending flood predictions based on current water depth. This prediction-sending witness task rewarded organisms that developed models that contained an execution path that checked the water depth, calculated a prediction, and transmitted that prediction. The associated property task only rewarded a model if every possible execution path performed that same behavior. Checking if a model supported a scenario was simple and quick, however checking if a model satisfied a witness or property task was difficult and time-intensive; in the worst case all possible execution paths of the model had to be checked.

To avoid unnecessary witness and property task checking, models were required to support a minimum set of scenarios before they were even considered as candidates for satisfying overall system properties. For example, a model could not

perform the previous witness/property example of sending a prediction based on current water depth if it did not use some method to check the water depth and successfully send its prediction. Thus, there was no reason to check for this system property unless a model supported one scenario associated with each of those behaviors. In fact, to satisfy any of the Grid-Stix behavioral requirements, a model needed to support one of each of the scenario alternatives (i.e., one processor speed and one transmission method), as well as 3 other required scenarios. These combinations of the 3 processor speed scenarios and 6 transmission method scenarios yielded 18 possible behavioral models or phenotypes, each of which represented a different combination of the non-functional properties (energy efficiency, prediction accuracy, and fault-tolerance). Although the previous Avida-MDE study successfully generated satisfactory behavioral models that represented some of the phenotypes, diverse models were found only by evolving many separate populations (the original study evolved 40 separate populations each with 3,600 individuals), and still the experiments were unable to discover all 18.

Experiments and Results

Generating a diverse suite of models

Our first objective is to assess how well the Eco-EA version of Avida-MDE performs compared to the original, single-niche version of Avida-MDE. The Grid-Stix problem provides an excellent case study for comparison, since one of the desired outcomes is to generate a suite of models, each of which minimally satisfies the required properties specified by the developer, but may also contain additional behavior that makes it suitable for domains that were not explicitly provided. A simple way to determine what additional behavior a model may possess is to consider which scenario it uses from each of the non-functional scenario sets. As described previously, there are 18 possible combinations of NF scenarios and therefore 18 unique phenotypes a model may represent, each of which yields a slightly different behavior in terms of energy efficiency, prediction accuracy, and fault-tolerance. The original version of Avida-MDE was unable to evolve all 18 possible phenotypes, even across 40 runs.

We compare the efficacy of the Eco-EA version of Avida-MDE in evolving a diverse suite of models to Goldsby and Cheng’s previous results (Goldsby and Cheng 2008b). The key difference between the two approaches is how the NF scenarios are rewarded. In both versions of Avida-MDE, organisms can only receive a fitness gain for one scenario from each of the sets of NF scenarios (in the Grid-Stix study, one processor speed and one transmission method). If an organism supports multiple scenarios from a given set, then it is rewarded only for the first one it supports. In the original Avida-MDE, all tasks in the environment, including these scenario tasks, add a fixed amount to an organism’s fitness when they are performed. In the Eco-EA version, each NF scenario task corresponds to a limited resource in the environment. When an organism performs one of these scenario tasks it consumes a fraction of the available resource, reducing the amount of that resource available to other

organisms. The fitness gain the organism receives is proportional to the amount of resource it consumes. This resource-dependent fitness encourages organisms to evolve to support little-used scenarios, and creates an overall diverse population of models in terms of non-functional properties. The rest of the Avida-MDE tasks (including the required scenarios) are still rewarded in the Eco-EA using the standard fixed-reward method; these tasks represent properties and behavior required in all models and therefore we want them to confer a constant fitness gain regardless of the number of other individuals performing the same tasks.

We perform 2 sets of 20 experiments, one set in each version of Avida-MDE. Slight improvements made to the original Avida-MDE after the previous results were published necessitated re-running the initial experiments in order to fairly compare the results of the Eco-EA version of Avida-MDE. We ran each experiment for 25,000 updates (updates are units of time in Avida that are roughly proportional to generations) or 24 hours, whichever came first.

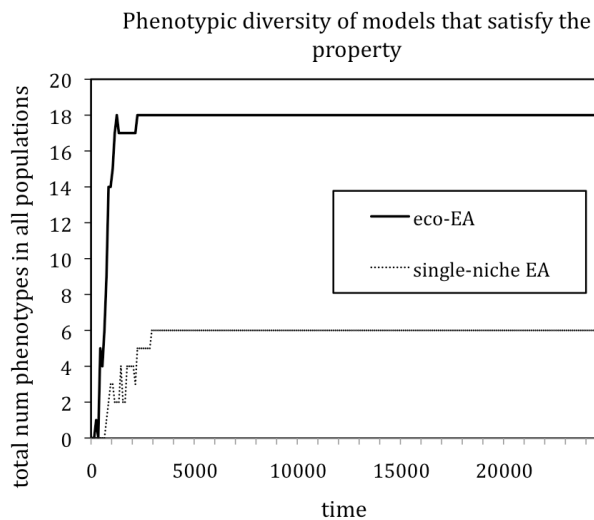


Figure 1. The number of unique phenotypes of models that satisfy the property in terms of non-functional property trade-offs found by all 20 runs in each environment over time. In this Grid-Stix problem there are 18 possible combinations of transition each of which results in different non-functional behavior in the models. In the Eco-EA (limited resource environment), invariant-satisfying models representing each of the 18 non-functional phenotypic possibilities quickly evolve. In the tradition EA (single niche environment), models satisfying the invariant evolve more slowly and fewer of the non-functional based phenotypes are found even after a long period of evolution. Each experiment evolves a population of 1,000 individuals for 24 hours or 25,000 updates, whichever comes first.

As discussed above, checking property and witness tasks is time-consuming, leading populations to become very slow in Avida time once many individuals satisfy the requirements to be checked for these tasks, so the absolute 24 hour time limit

is imposed as well. In this pair of experiments all of the 20 Eco-EA replicates evolve to satisfy the property task and reach the 24 hour limit, ending between 1,000 and 5,000 updates. Ten of the single-niche EA replicates reach the 24 hour limit (the 9 that evolve the property task and one other that has models being checked for the property though it never evolves), ending between 2,000 and 23,000 updates, and the other 10 end at the 25,000 update cutoff.

We find that the Eco-EA version of Avida-MDE not only generates a more diverse suite of final model phenotypes, but that it also evolves models satisfying the required functional property significantly faster than the traditional, single-resource approach. Figure 1 shows the number of total unique phenotypes of models satisfying the required property found across 20 Avida experiments over time. The Eco-EA finds models satisfying the property before reaching 1,000 updates of evolution (~400 generations), and all 20 replicates find models by 5,000 updates. Across all 20 replicates the Eco-EA finds property-satisfying models of each of the 18 non-functional phenotypes within 2000 updates of evolution (~800 generations). In contrast, the traditional approach using a single niche only finds any model satisfying the required property in half of the replicates, and even in those that do find a satisfactory model the average time one is found is three times as long as in the Eco-EA (5000 updates vs. 1500 updates). Even after 25,000 updates of evolution the single niche approach finds property-satisfying models representing only 6 of the 18 possible phenotypes.

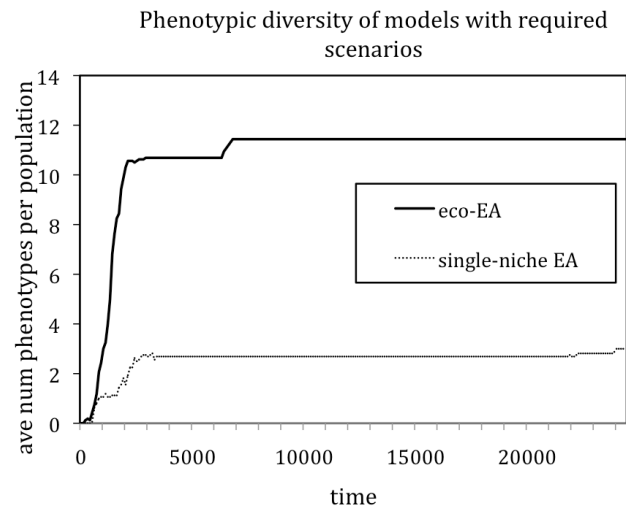


Figure 2. The average number of unique phenotypes of all models in each population in terms of non-functional properties. Eco-EA populations quickly diversify to cover most of the possible phenotypes well before evolving models that satisfy the property, while the single-niche EA is stuck on just one or two phenotypes per population. This means there are less evolutionary paths to find a model satisfying the property in the single-niche EA, and hence it takes longer. Each experiment evolves a population of 1,000 individuals for 24 hours or 25,000 updates, whichever comes first.

The Eco-EA version of Avida-MDE also yields a significantly more diverse set of models in each individual experiment than the single-niche EA. Every one of the 20 experiments using the Eco-EA yielded property-satisfying models. The final populations contained coexisting models representing between 8 and all 18 different phenotypes, with a mean of 14.8 phenotypes per population. In contrast, only 9 of the 20 single-niche Avida-MDE experiments evolved any property-satisfying models, with a maximum of 4 phenotypes in a single population. The average number of phenotypes found in the final populations of single-niche EA experiments was 2.85 ($p < .001$ comparing 2.85, $s = 3.7$ to 14.8, $s = 2.9$, with 38df, using the independent group t-test for means).

One could argue that since we know all 18 target phenotypes, we could simply evolve each of them in independent populations. However, there are several reasons we would expect this seemingly simpler method would not perform as well as Eco-EA. First, the Eco-EA is more generalizable to other problems; in many cases, developers will not know *a priori* what novel behavior a model may evolve and thus it is not always possible to enumerate the desired phenotypes. Second, the complex behavior required for a model to satisfy the required functional properties must be built on simpler behavior such as supporting scenarios. We posit that rewarding for many scenarios yields more potential pathways for evolution to follow in finding a model that satisfies the property.

Once a single property-satisfying model is found, it may be possible for that model to change its non-functional behavior while still maintaining the required behavior.

The theory that the inclusion of more scenarios yields more evolutionary pathways and thus leads to faster evolution also may explain why the Eco-EA finds models satisfying the developer's requirements faster than the single-niche EA. Figure 2 shows the average number of unique phenotypes (based on NF scenarios) of all models in each population, including those that do not satisfy the required property. To test this theory we performed experiments where instead of including tasks for all of the NF scenarios in the environment, we included only one scenario from each of the 2 sets, a single processor speed and a single transmission method. We performed 5 replicates of each of the 18 environments thus created, for a total of 90 experiments (as compared to the 20 performed including all of the scenarios). We found that when only rewarding for a single phenotype, no model satisfying the required behavioral property ever appeared. The Eco-EA populations diversify quickly to contain individuals of almost all of the phenotypes in each population, while the single-niche populations are stuck on just one or two of the possible phenotypes, giving evolution fewer possible paths to a model satisfying the property.

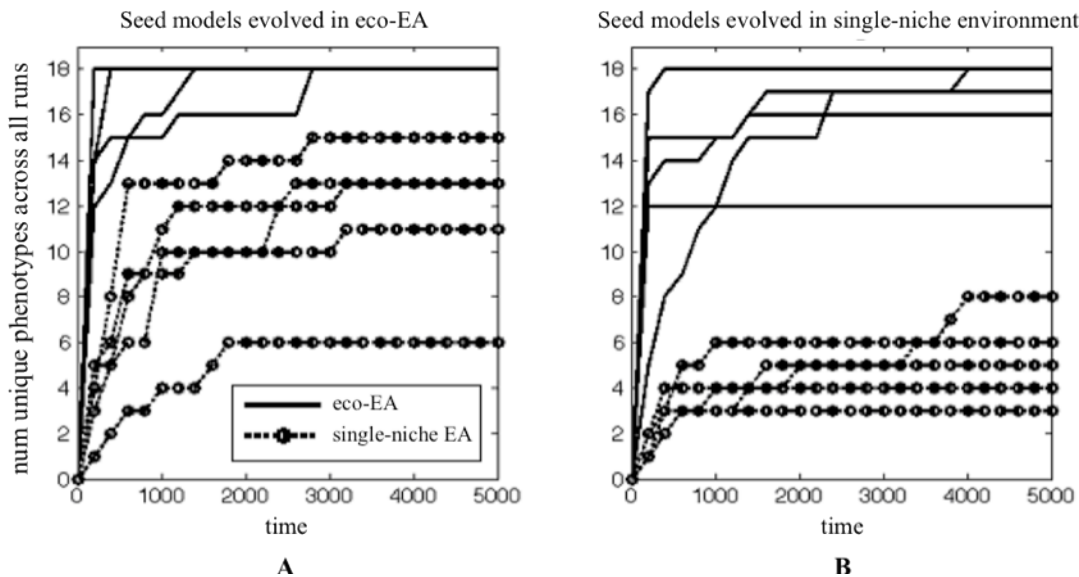


Figure 3. The number of unique phenotypes of models that satisfy the property found by all 20 runs for each treatment over time. (A) Performance of each version of Avida-MDE when seeded with each of the 5 models originally evolved in the Eco-EA environment. While the individual models yield highly varying results, the Eco-EA quickly evolves all 18 phenotypes no matter which of the 5 it is seeded with. The single-niche environment is never able to find all 18 phenotypes. (B) Similar results occur when populations are seeded with models originally evolved in the single-niche environment. The Eco-EA now only generates all 18 phenotypes for 2 of the initial models, but still generates more phenotypes in the worst case (12) than the single-niche EA generates in the best case (8). Each experiment evolves a population of 1,000 individuals for 24 hours.

Evolvability of Models

A common situation is for a developer to have already developed one model suited to a given set of conditions, and needs a suite of models appropriate for a variety of condition domains. We therefore compared the evolving population of the Eco-EA version of Avida-MDE to that of the single-niche EA when the population is initially filled with copies of one individual that builds a model already satisfying the required behavior.

We randomly selected 5 individuals that generated models satisfying the required property from those evolved using the Eco-EA version of Avida-MDE, with the specification that they each come from a different replicate population and each represent a different non-functional phenotype. We then did the same with the models evolved using the original Avida-MDE, ensuring that we chose the same 5 phenotypes as the former set. For each of the 10 chosen models, we used the model to seed the initial populations of 20 replicate experiments where we continued evolution in the Eco-EA environment, and 20 where we continued evolution in the original single-niche environment.

We find two key results; 1) the Eco-EA environment generates a more diverse suite of models more quickly than the original single-niche environment; 2) the individuals evolved in the Eco-EA environment appear to be more evolvable in terms of generating diverse phenotypes than those evolved in the single-niche environment. Figure 3 shows that the Eco-EA version of Avida-MDE quickly generates diverse populations representing models of many (and often all) phenotypes no matter which model the population is seeded with, while the single-resource EA tends to only evolve phenotypes close in genetic space to that of the initial model.

It also appears that models originally evolved in the Eco-EA environment yield more diverse phenotypes in either environment when they are used to seed the initial population; the Eco-EA generates all 18 possible phenotypes when seeded with any of the 5 models initially evolved using the Eco-EA, and the single-niche EA generates over 11 phenotypes when seeded with 4 of these models, while the most it ever finds when seeded with models initially evolved in the single-niche environment is 8 phenotypes. The increased evolvability of models initially evolved in the Eco-EA version of Avida-MDE can be seen more clearly in figure 4, where the average results across all 5 seed models are shown for each of the 4 treatments.

Once again we find that the Eco-EA version of Avida-MDE not only evolves a more diverse set of phenotypes more quickly than the single-resource approach across sets of all 20 runs, but it also yields higher diversity in individual runs. When averaging all runs across all 10 seed models, the Eco-EA evolves an average of 17.1 phenotypes per run, while the single-resource EA evolves an average of only 8.4 phenotypes ($p < .001$ comparing 17.1, $s = 1.25$ to 8.4, $s = 2.7$ using the independent group t-test for means).

The individual run diversity also differs based on which environment the seed models were evolved in. Averaging all runs from both environments when seeded with the 10 models evolved using the Eco-EA, 14.8 unique phenotypes are generated per run, vs. 10.7 phenotypes per run when populations are seeded with the models evolved in the single-

niche environment ($p < .001$ comparing 14.8, $s = 1.7$ to 10.7, $s = 2.2$ using the independent group t-test for means).

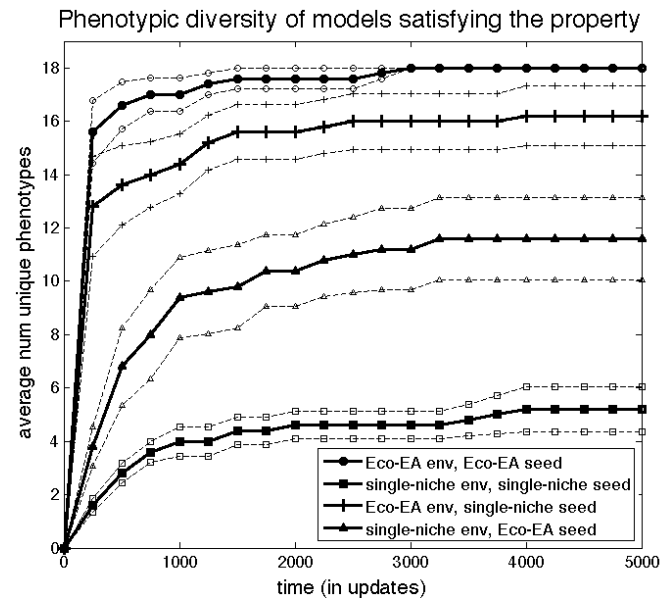


Figure 4. Average of data with error bars (± 1 standard error) for each of 4 experimental treatments (All combinations of 2 types of seed models; those evolved in the Eco-EA environment or those evolved in the single-niche environment, and 2 environments for continued evolution; the Eco-EA and the single-niche). The line for each treatment represents the average of the 5 sets of experiments, one for each model used in that treatment. The data for each of the 5 sets is the number of unique phenotypes found by all 20 populations in that set over time. The Eco-EA finds on average a more diverse set of models than the single-niche EA no matter which type of models it is seeded with. Both environments find a significantly more diverse set of models when seeded with models initially evolved using the Eco-EA than those evolved using the single-niche EA.

This result is something we would like to explore more thoroughly, as we can not yet identify what exactly makes the models evolved in the Eco-EA environment more able to diversify in any environment during further evolution. We found that one of the models evolved by the Eco-EA actually represented multiple phenotypes itself, as it stochastically performed one of 2 different options for the transmission scenario. However the other 4 models did not show this behavior so that cannot explain the overall result. Hypotheses we would like to test include that the Eco-EA evolved models could do well when they switch frequently between performing different scenarios, and so there may be selective pressure for them to be only one or two mutations away from performing a different set of scenarios at any given time.

Conclusion

In this paper, we compared the performance of Eco-EA to a more traditional EA (Avida-MDE) on a complex software engineering problem. Specifically, we used both Eco-EA and Avida-MDE to generate software models for a flood warning system. For this problem, there were 18 possible models (phenotypes) that all met the functional system objectives (i.e., detect flooding), but did so using a variety of different non-functional tradeoffs. Eco-EA provided three significant advantages over Avida-MDE. First, Eco-EA more rapidly evolved organisms that generated models that satisfied the developer's requirements. Second, Eco-EA evolved a more diverse set of solutions that represented models with different properties. Lastly, when the models created by Avida-MDE and Eco-EA were used as seeds for subsequent experiments, the solutions created by Eco-EA exhibited greater evolvability. These results indicate that the Eco-EA facilitates the evolution of solutions to complex problems.

In the future, we plan to apply Eco-EA to complex problems in different domains. One potentially interesting area of investigation is problems whose solutions may require explicit cooperation among the various species present within the population. Additionally, we are working on extending Eco-EA to other areas of evolutionary computation, such as natural problem decomposition and multi-objective optimization.

Acknowledgements

This work has been supported in part by NSF grants CCF-0541131, CCF-0820220, CCF-643952 and CCF-0523449, DBI-0939454, Army Research Office grant W911NF-08-1-0495, Ford Motor Company, the DARPA "Fun Bio" program (HR0011-05-1-0057), and a Quality Fund Program grant from Michigan State University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, U.S. Army, Ford, or other research sponsors. The authors also thank C. Adami and everyone in the Digital Evolution Lab at Michigan State for their support, ideas, and critiques contributed to this research, especially Dave Knoester.

References

- Cooper, T.F. and Ofria, C. (2002). Evolution of stable ecosystems in populations of Digital Organisms. In *Artificial Life VIII*, pages 227-232. MIT Press, Cambridge, MA.
- Tyson, G.W. et al. (2004). Community structure and metabolism through reconstruction of microbial genomes from the environment. *Nature*, 428:37-43.
- Gaston, K. J. (2000). Global Patterns in Biodiversity. *Nature* 405:220-227.
- Tilman, D. (1982). *Resource Competition and Community Structure*. Princeton University Press, Princeton, NJ.
- Goings, S. and Ofria, C. (2009). Ecological Approaches to Diversity Maintenance in Evolutionary Algorithms. In *IEEE-Alife*, pages 124-130. Published by IEEE.

- Ofria, C. and Wilke, C.O. (2004). Avida: A Software Platform for Research in Computational Evolutionary Biology. *Artificial Life*, 10:191-229.
- Goldsby, H. J. and Cheng, B. H. C. (2008a). Automatically Generating Behavioral Models of Adaptive Systems to Address Uncertainty. In *ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MoDELS)*, pages 568-583. Springer-Verlag Berlin, Heidelberg.
- Schmidt, D.C. (2006). Model-Driven Engineering. *IEEE Computer*, 39:2.
- Goldsby, H. J. and Cheng, B. H. C. (2008b). Avida-MDE: A Digital Evolution Approach to Generating Models of Adaptive System Behavior. In *Genetic and Evolutionary Computation Conference*, pages 1751-1758. ACM, New York City, NY.
- Hughes, D., Greenwood, P., Coulson, G., Blair, G., Pappenberger, F., Smith, P. and Beven, K. (2006). An intelligent and adaptable flood monitoring and warning system. In *5th UK E-Science All Hands Meeting*.
- Konrad, S., Goldsby, H., and Cheng, B. H. C. (2007). i2MAP: An Incremental and Iterative Modeling and Analysis Process. In *ACM/IEEE Int. Conference Model-Driven Engineering Languages and Systems*, pages 451-466. Springer, New York City, NY.

The Evolution of Temporal Polyethism

Heather J. Goldsby¹, Neem Serra¹, Fred Dyer², Benjamin Kerr¹, Charles Ofria²

¹University of Washington, Seattle, WA 98195

²Michigan State University, East Lansing, MI 48824

Abstract

Temporal polyethism is a method of division of labor exhibited by many eusocial insect colonies, where the type of task an individual attempts is correlated with its age. The evolutionary pressures that give rise to this widely-observed pattern are still not fully known. The long generation times of eusocial insects combined with the complications associated with performing artificial selection experiments on colonies of organisms makes this topic challenging to investigate using organic systems. In this paper, we use digital evolution to explore whether temporal polyethism may result from pressures to preserve colony members in the face of varying degrees of risk associated with different tasks. Specifically, we require a colony of digital organisms to repeatedly perform a set of tasks in order for the colony to replicate. We associate the different tasks with different lethality risks. Under these conditions, we observe that the digital organisms evolve to perform the less risky tasks earlier in their life and more risky tasks later in life, regardless of the order in which the tasks were performed by the ancestor organism at the start of the experiment. These results demonstrate that pressures resulting from the relative riskiness of various tasks and aging is sufficient to favor the evolution of temporal polyethism.

Introduction

Division of labor, where individuals specialize on specific roles and cooperate to survive, is hailed as a strategy central to the success of eusocial insect, crustacean, and mammal colonies (Crespi, 2001; Duffy, 2003; Hölldobler and Wilson, 2009; Jandt and Dornhaus, 2009; Queller and Strassmann, 2003; Wilson, 1980). Within nature, eusocial organisms are renowned for exhibiting *reproductive division of labor*, where members of the reproductive caste (i.e., queens) produce offspring and members of the non-reproductive caste care for the brood and perform other duties central to the maintenance of the eusocial colony (Jandt and Dornhaus, 2009). Moreover, many eusocial organisms, such as leaf-cutter ants (Wilson, 1980), bumblebees (Jandt and Dornhaus, 2009), and aphids (Pike and Foster, 2008), also exhibit *task-related division of labor*, where individuals specialize on performing a particular task. For example, non-reproductive worker bumblebees specialize to perform roles

that include foraging, caring for the brood, building honeypots, guarding the hive, or cooling the hive through fanning (Jandt and Dornhaus, 2009).

One form of task-related division of labor exhibited by many eusocial colonies is *temporal polyethism*, where a worker's age is correlated with the type of task it performs (Franks et al., 1997; Hölldobler and Wilson, 2009; Robson and Beshers, 1997; Sendova-Franks and Franks, 1993; Tofilski, 2002; Tofts, 1993; Traniello and Rosengaus, 1997). For example, within a honeybee colony, a worker bee may progress sequentially through four castes: cell cleaning caste, broodnest caste, food storage caste, and forager caste (Seeley, 1982). Within ant colonies, a similar shift is performed from activities within the nest, such as brood care, to foraging activities outside the nest (Hölldobler and Wilson, 2009). Researchers are still actively exploring the causes and mechanisms underlying this division of labor pattern. In this paper, we study the evolutionary conditions that can give rise to temporal polyethism.

Two hypotheses have been proposed to explain temporal polyethism. The *task-riskiness* hypothesis posits that an individual's age is causally linked to the task that it performs (Hölldobler and Wilson, 2009; Robson and Beshers, 1997; Traniello and Rosengaus, 1997). This causal relationship is thought to have evolved because of a pressure to conserve work force members and thus to have older members (who are closer to death) perform more risky tasks (Hölldobler and Wilson, 2009). For example, foraging, a task commonly responsible for the loss of 1% to 10% of the colony population per day (Hölldobler and Wilson, 2009), is performed when the organism is likely to die of natural age-related causes and thus is more expendable. In this way, the colony optimizes the use of its workers. (Tofilski, 2002). In contrast, the *foraging for work* hypothesis assumes that as organisms are born they perform tasks closest to them and proceed to perform tasks further from the center of the nest (Franks et al., 1997; Hölldobler and Wilson, 2009; Sendova-Franks and Franks, 1993; Tofts, 1993). This explanation depends only upon organisms' reactive responses to task stimuli. Thus, according to the foraging

for work hypothesis, colonies exhibit a temporal polyethism pattern as a result of the spatial organization of the colony's nest without any inherent evolutionary advantage related to the riskiness of any task.

Studies have produced evidence in support of both hypotheses (Franks et al., 1997; Hölldobler and Wilson, 2009; Robson and Beshers, 1997; Traniello and Rosengaus, 1997). Specifically, studies with monomorphic ants provide support for the foraging for work hypothesis by presenting evidence that the task riskiness hypothesis is too rigid to account for the unstable situation of ants and any correlation of age and task is merely a byproduct (Sendova-Franks and Franks, 1993). In the original foraging for work mathematical model created by Tofts, ants change tasks when work was unavailable at the current location (Tofts, 1993). In one study, marking the ants showed that older ants were flexible in the tasks they performed, and all ants, regardless of age, foraged for work, meaning that they actively sought out tasks to perform (Sendova-Franks and Franks, 1993). However, critiques of Tofts' model of foraging for work highlight that the way in which workers can move between tasks creates a biologically unrealistic colony (Robson and Beshers, 1997). Others have noted that Tofts' model does not account for many other eusocial insects, such as termites, that have a well-developed age-based division of labor strategy that is not a byproduct of foraging for work (Traniello and Rosengaus, 1997). In addition, an alternative mathematical model testing the task-riskiness hypothesis was created with a set of two tasks that each had a different mortality rate (Tofilski, 2002). This model shows that the longevity of workers in a colony that perform tasks without regard to the amount of risk associated with them is significantly lower than the longevity of workers a colony that perform tasks in order of risk (Tofilski, 2002).

While these studies have examined potential proximate causes of temporal polyethism exhibited by current eusocial colonies, it is challenging to explore the evolutionary conditions that may give rise to this pattern. Both field observations and experimental studies of evolution in lineages of actual organisms are infeasible because of long generation times and the complexity of studying large social groups in a controlled way.

To address these challenges, we use Avida, a digital evolution software platform that maintains a population of self-replicating computer programs in a user-defined environment (Ofria and Wilke, 2004). Each computer program is a digital organism that executes its genome (a list of computer instructions) to perform tasks, where the tasks enable the organism to collect resources and thus compete with its neighbors. Avida meets all of the requirements for evolution: replication, variation, and differential selection. Avida has previously been used to study topics such as division of labor (Goldsby et al., 2012), origin of complex features (Lenski et al., 2003), and evolution of cooperation (Knoester et al.,

2007). Digital organisms have rapid generation times (e.g., thousands of generations in a few hours), thus enabling us to study this complex evolutionary phenomenon.

In this paper, we use Avida to explore whether varying the amount of risk associated with tasks is sufficient to evolve colonies that exhibit a temporal polyethism structure. We created a world in which different tasks were associated with different levels of risk. We used colonies of clonal (i.e., genetically identical) organisms, where the colonies competed for limited space in the Avida world. Each colony was required to perform each type of task a certain number of times for the colony to replicate. An ancestor organism performed each of the required tasks once. We explicitly removed any spatial component to task performance to determine whether organisms were responding to the spatial structure of the nest, or the risk associated with tasks. In response to these pressures, the organisms evolved division of labor strategies in which tasks associated with less risk were done earlier in an organism's life and riskier tasks were performed later in life, regardless of the initial order of the tasks. These data provide support for the hypothesis that risks associated with aging and various tasks are sufficient to produce temporal polyethism.

Methods

To use Avida to study the evolution of temporal polyethism, we created a world consisting of competing colonies that each contain a set of clonal organisms. Each of digital organisms has a virtual CPU, a genome (a circular list of computer instructions), and a location within the colony. The virtual CPU of an organism consists of three general-purpose registers and two stacks. Each digital organism executes instructions on its virtual CPU. The instruction set in Avida allows for basic computational tasks, such as addition, multiplication, and bit-shifts, controlling the execution flow, and self-replication. An organism performs logic operations (NOT, NAND, etc.) called *tasks* by executing the instructions in their genome.

For a colony to replicate, the organisms within that colony must perform each type of task in a set a certain number of times. For example, in our initial experiments, a colony had to perform task NOT 250 times and task NAND 250 times. A natural analog is a colony of eusocial insects in which the workers must both forage for food and tend to the brood. In addition, because each colony starts with only one organism, organisms must also replicate to produce other organisms that can assist them in the performance of tasks to achieve the overall colony objective. During colony replication, the genome of the colony is potentially mutated (i.e., instructions are potentially inserted, removed, or exchanged for other instructions). This new genome is used to seed a daughter colony, which is selected randomly from the colony population.

To address our central question regarding the evolution of

temporal polyethism, we added the capability for each task to be associated with a *lethality risk* that specifies the probability of the organism dying before completing the task. Non-risky (or safe) tasks have a lethality risk of 0. Our most risky tasks have a lethality risk of 25%. If an organism is killed while performing a task, then the task is not completed and thus does not count toward the task count of the colony.

In most other Avida experiments, organisms are reset upon producing an offspring, in order to emulate the behavior of bacteria that divide into two daughter cells when they replicate. However, since age and internal state play a key role in these experiments, we modified the organisms so that they do not reset after replication, but rather just continue running.

At the outset of these experiments, we seed the colonies with an *ancestor* organism that performs all the types of tasks necessary for completion of the colony task. In our experiment, an ancestor organism performs task NOT and task NAND once. Because each colony contains only one individual at the onset of the experiment and also after colony replication, organisms must self replicate to fill the colony. Each experiment comprises several different treatments that randomize the order in which the tasks appear in the ancestor organisms' genomes, as well as the riskiness associated with the tasks.

The starting world for each experiment had 400 colonies each of which contained one ancestor organism. Organisms were subject to three mutation rates during colony reproduction: a copy mutation rate of 0.0075 (0.0003 per instruction), an insertion mutation rate of 0.05 (0.002 per replication), and a deletion mutation rate of 0.05 (0.002 per replication). For each experiment, we conducted 30 trials to account for the stochastic nature of evolution. Each trial ran for 100,000 updates, where an *update* is the amount of time it takes an average organism to execute 30 *cycles* – each instruction takes one cycle to execute.

Results

The primary topic of this study is whether the risks associated with aging and tasks are sufficient to evolve colonies of organisms that exhibit temporal polyethism. For our study, we created a two-task environment in which colonies had to perform task NOT 250 times and task NAND 250 times in order for the colony to replicate. We created four risk treatments (described in Table 1) that vary the lethality risks associated with the tasks. Specifically, the treatments are: (1) task NOT is risky, (2) task NAND is risky, (3) neither task is risky (a control), and (4) both task NOT and task NAND are risky (a control).

Additionally, we created two possible ancestor organisms (depicted in Figure 1). Each ancestor completes each task once and then self-replicates. However, ancestor NOT-NAND performs the NOT task first and ancestor NAND-NOT performs the NAND task first. While we depict the tasks as

Task	Risk Treatments			
	NOT risky	NAND risky	No risk	Both risky
NOT	25%	0%	0%	25%
NAND	0%	25%	0%	25%

Table 1: The four risk treatments for a two-task environment. The rows describe the lethality risks associated with tasks NOT and NAND. (E.g., A 25% risk means that while performing the task, the organism has a 25% chance of dying.) The columns describe a specific treatment.

atomic units within this Figure to denote order, to actually perform a task an organism must execute several instructions. By varying the ancestor organism, we are able to verify that any patterns of temporal polyethism result from the riskiness associated with the tasks, not the initial genomic structure of the organisms. For each ancestor, we performed all four risk treatments. If task riskiness is a sufficient pressure to result in temporal polyethism, then we should see that organisms evolve to perform the less risky task first and the more risky task second, regardless of whether NOT or NAND is the risky task, and the initial order of the tasks with the ancestor organism's genome.

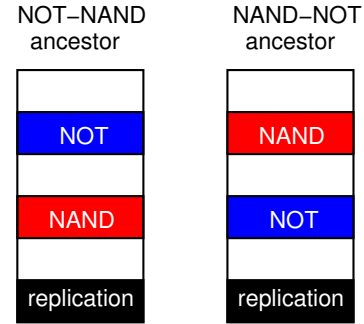
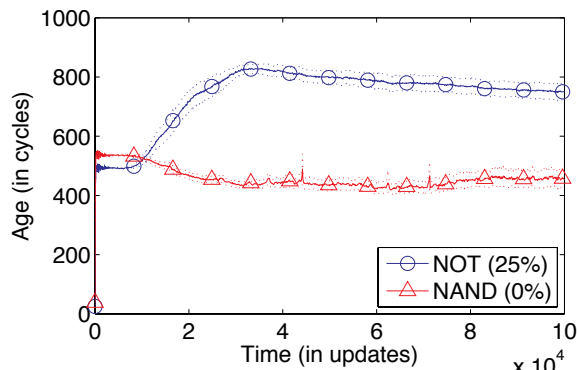


Figure 1: The layout of the ancestor organisms for two-task temporal polyethism experiments. The NOT-NAND ancestor performs task NOT, performs task NAND, and then replicates. The NAND-NOT ancestor performs task NAND, performs task NOT, and then replicates. Because the genomes are circular, after each organism replicates, it resumes execution at the top of its genome.

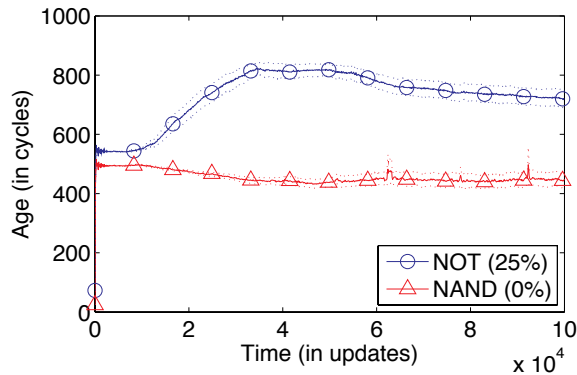
Figures 2 and 3 depict the results of the experimental treatments. For all results, the mean age at which a task is performed includes the age of organisms who died attempting to perform that task. Figure 2 depicts the treatments in which task NOT is risky. In both treatments that vary the ancestor organism, the mean age at which NOT is performed is significantly greater than the mean age at which NAND is performed (Mann-Whitney U Test). For example, for the NOT-NAND ancestor, NOT is performed at the mean age of 750.37 ± 27.45 cycles and NAND is performed at the mean age of 453.43 ± 29.12 cycles. The treatment seeded with the

NOT-NAND ancestor reversed the order in which the tasks were performed in 26 out of 30 replicates. Additionally, 23 out of 30 replicates seeded with the NAND-NOT ancestor performed the riskier task NOT at a later age than task NAND.

Figure 3 depicts the treatments where task NAND is risky. For both treatments, the mean age at which NAND is performed is significantly greater than the mean age at which NOT is performed (Mann-Whitney U test). 27 out of 30 replicates with the NOT-NAND ancestor and 28 out of 30 replicates with the NAND-NOT ancestor performed the riskier task NAND at a later age than task NOT. These treatments support our hypothesis that task riskiness can result in temporal polyethism in which the more risky task is performed later in the lifetime of the organisms.



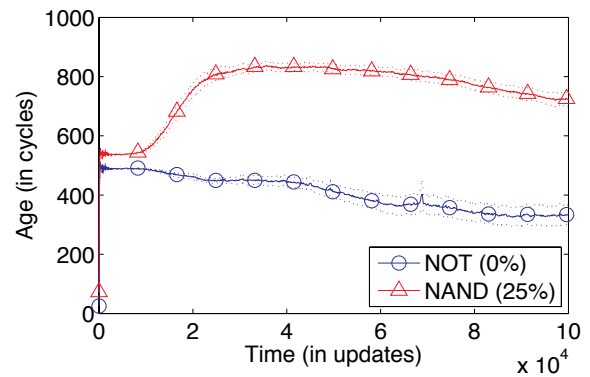
(a) Ancestor: NOT-NAND; Treatment: NOT is risky



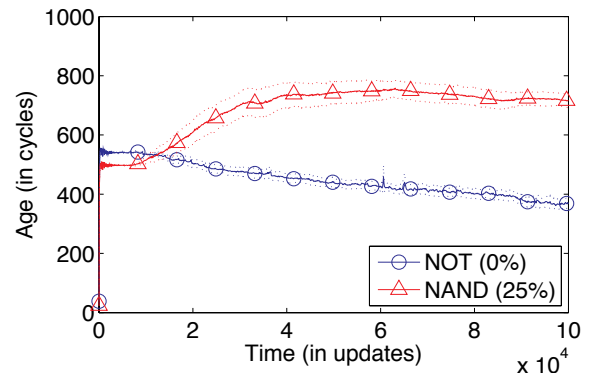
(b) Ancestor: NAND-NOT; Treatment: NOT is risky

Figure 2: Task ordering over time in treatments where task NOT is risky compared across different ancestors. For each plot, the x-axis is evolutionary time and the y-axis is the mean age in cycles when the associated task is performed. Dotted lines represent standard error. Task NOT is consistently performed later in the lifetime of the organisms, regardless of the starting order.

Figures 4 and 5 depict the results of our controls, which are designed to verify that, given the same level of risk, there is nothing inherent in the tasks that results in one being per-



(a) Ancestor: NOT-NAND; Treatment: NAND is risky



(b) Ancestor: NAND-NOT; Treatment: NAND is risky

Figure 3: Task ordering over time in treatments where task NAND is risky compared across different ancestors. For each plot, the x-axis is evolutionary time and the y-axis is the mean age in cycles when the associated task is performed. Dotted lines represent standard error. Task NAND is consistently performed later in the lifetime of the organisms, regardless of the starting order.

formed earlier or later in the organisms' lifetimes. Figure 4 depicts the results of the control treatments in which neither task is risky. For these control treatments, the average age at which organisms perform tasks increases over the duration of the experiment. This change results from individual organisms evolving to perform the same task multiple times within their lifetime resulting in the average age of task performance increasing. However, the mean age at which task NOT is performed is not significantly different than the mean age at which task NAND is performed (Mann-Whitney U Test). Figure 5 depicts the results of the control treatments in which both tasks are risky. For both treatments, the mean age at which the organisms perform the tasks reflects their order in the genome. One thing to note about this control is that the high level of risk associated with both tasks decreases the rate of colony replication. In fact, many colonies lost the ability to replicate altogether and survived merely because other colonies within their trial were also unable

to replicate. Thus, these colonies are not actually evolving in an adaptive fashion. However, the data provided by the controls indicate that there is nothing inherent in the NOT or NAND tasks that implies an ordering. Taken together, these treatments indicate that more risky tasks are, on average, performed later within the lifetime of the organisms.

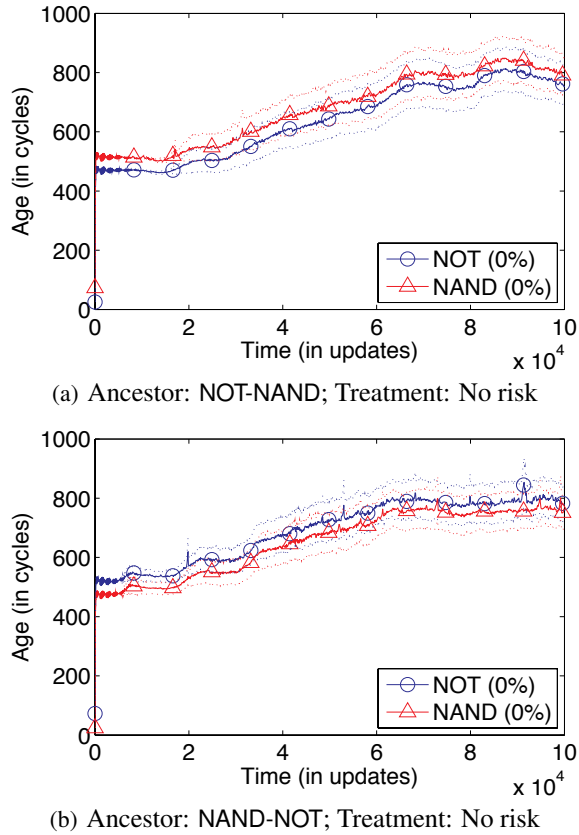


Figure 4: Task ordering over time in control treatments where neither task is risky. For each plot, the x-axis is evolutionary time and the y-axis is the mean age in cycles when the associated task is performed. Dotted lines represent standard error. In these results, the controls indicate that there is nothing intrinsic about the tasks that is driving the temporal polyethism results.

To better understand how the colonies were responding to the amount of risk associated with a task, we performed several additional treatments in which we set the lethality risk for the risky task to 7%, 15%, and 20%. For these new risk conditions, we again varied the ancestor and also which task was risky. Figure 6 shows the number of replicates out of 30 that evolved a temporal polyethism pattern, where the more risky task was performed later in life. For all risk levels, if the ancestor organism had properly ordered the tasks (i.e., it performed the risky task last), then most replicates were able to maintain the temporal polyethism pattern. For example, when NOT is the risky task, most replicates with

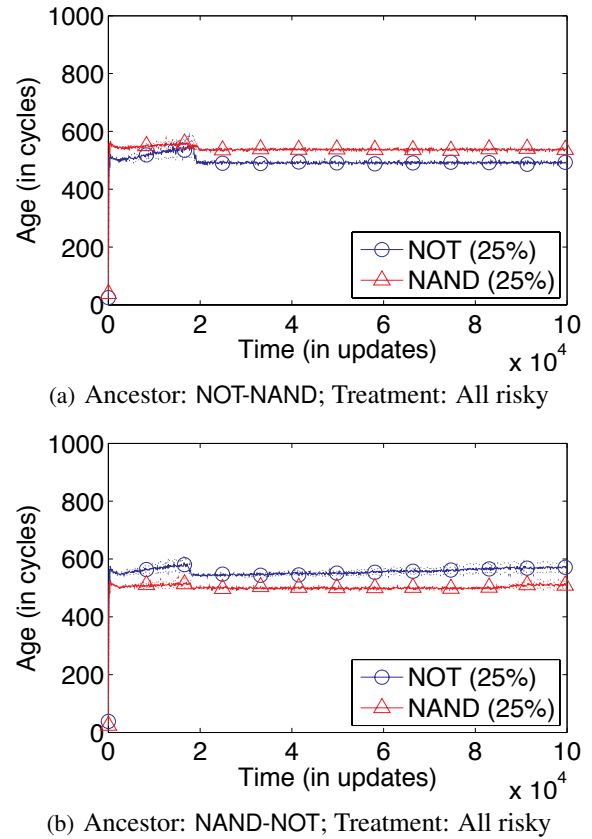
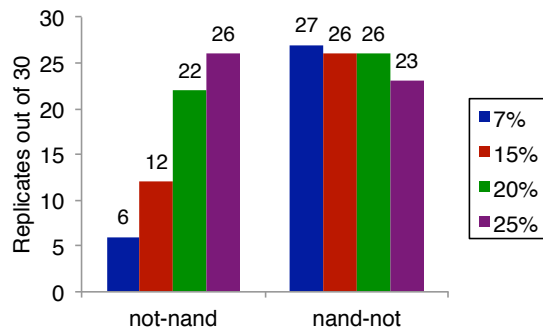


Figure 5: Task ordering over time in control treatments where both tasks are risky. For each plot, the x-axis is evolutionary time and the y-axis is the mean age in cycles when the associated task is performed. Dotted lines represent standard error. In these results, the controls indicate that there is nothing intrinsic about the tasks that is driving the temporal polyethism results.

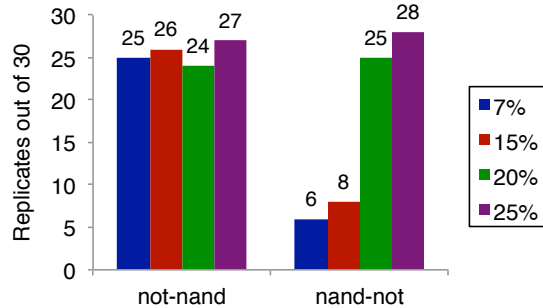
the ancestral organism NAND-NOT maintained the ordering present in the ancestor genome and performed NOT later in life. However, these data also reveal that at lower risk levels, fewer replicates were able to evolve the temporal polyethism pattern if the ancestral organism started with the riskier task being done earlier in life. For example, fewer replicates with the ancestral organism NOT-NAND were able to rearrange their genomes such that the risky task NOT was done later in life when the lethality risk was lower. These results indicate that the level of risk plays an important role in the evolution of temporal polyethism.

Analyses

We have demonstrated that colonies evolve to perform more risky tasks, on average, later within their lifetime than safe tasks. Next, we examine how this behavior interacts with reproduction and then conduct a case study analysis of a colony that exhibits this behavior.



(a) Treatment: NOT is risky



(b) Treatment: NAND is risky

Figure 6: The results of the temporal polyethism treatments, where risk level was varied. The y-axis of both plots is the number of replicates out of 30 that were able to do the risky task later in life. The x-axis shows the results from two different ancestors: NOT-NAND and NAND-NOT. (a) shows results from when NOT is the risky task and NAND does not have any risk. (b) shows results from when NAND is the risky task and NOT does not have any risk. The key denotes the lethality risk for the risky task.

Task Performance and Replication. Within these experiments, organisms have a pressure not just to perform tasks, but also to replicate and produce clones capable of performing these same tasks. One topic we were interested in exploring is when the organisms replicated. To address this topic, we examined a case study treatment from our original two-task experiment that begins with the NOT-NAND ancestor and in which task NOT is risky. Figure 7 depicts the mean age at which the tasks were performed and at which the organisms replicated. Intriguingly, the organisms performed the less risky task (NAND), replicated, and then much later in their life performed the more risky task (NOT). In this example, this result suggests that the organisms have evolved a strategy that balances their need to perform tasks, the risk associated with these tasks, and their need to replicate.

Two-Task Colony Case Study. Next, we examined the behavior of a successful colony from our two-task experi-

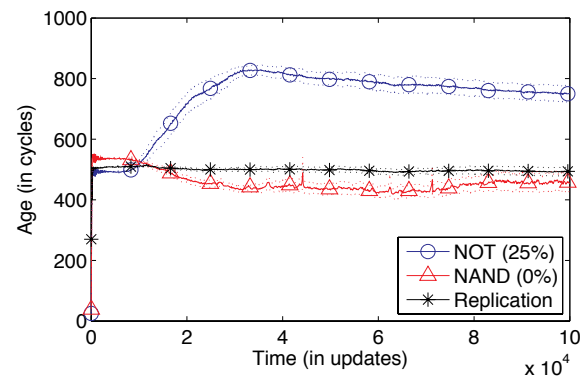


Figure 7: These results depict the mean age at which task NOT (blue line with circles), task NAND (red line with triangles) and replication (black line with stars) are performed for the case study treatment where NOT is risky and the runs were started with the NOT-NAND ancestor. These results suggest that the organisms are performing task NAND one or more times, replicating, and then performing task NOT.

ment that begins with the NOT-NAND ancestor and in which task NOT is risky to ascertain how it managed task performance and replication (results depicted in Figure 2a). The organisms within this colony executed a precise behavioral plan that is depicted in the phenotype portion of Figure 8. They performed task NAND, replicated, performed task NAND again, replicated again, and then repeatedly performed task NOT (the risky task) until it killed them. The organisms in this case study clearly exhibit the temporal polyethism pattern of performing the risky task after their other duties had been completed.

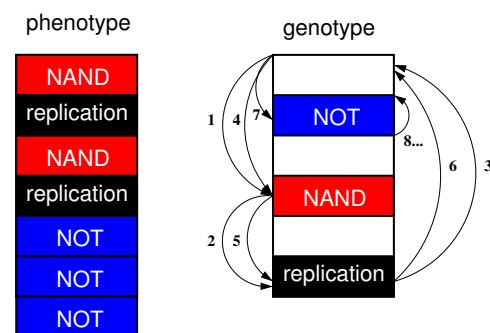


Figure 8: Diagrams of the phenotype (left) and genotype (right) of a case study organism whose colony exhibited temporal polyethism with two tasks. The numbered arrows surrounding the genotype indicate the order in which instructions were executed to produce the phenotype. In this case, the genotype is very similar to the NOT-NAND ancestor. The risk-based order in which the tasks were performed depended upon control-flow instructions in the genome.

A second topic we explored was how the genome archi-

ture of this case study supported this behavior. For example, organisms may have rearranged their genome to support task ordering (i.e., by moving the instructions that performed more risky task to the end of their genome) or organisms may have evolved to use control-flow instructions that enable them to skip over portions of their genome. In this case, the organisms evolved to use the control-flow instructions. The architecture of the genome, which is depicted in the genotype portion of Figure 8, is extremely similar to the ancestor organism: task NOT is encoded first, then task NAND, and lastly replication. However, the organisms evolved to have both jump instructions (to skip task NOT until the remainder of the genome had been executed twice) and a loop to continue to perform task NOT until death. Organisms set and used the value of a register that was preserved during replication to track which genome iteration they were on and to modify their behavior accordingly. The numbered arrows in Figure 8 depict the order in which the elements of the genome were executed.

Measuring Temporal Polyethism. There are two challenges associated with measuring temporal polyethism: First, each organism may perform each task multiple times over its lifetime. Second, an organism may die while performing a task as either the consequence of the lethality risk associated with that task or as the result of being replicated over by a neighboring organism. Thus far, to measure temporal polyethism, we have examined the mean age at which organisms perform a task. Here we assess this measurement by comparing it to two other potential measurements: (1) the mean age at which the organisms *first* perform a task, and (2) the mean age at which the organisms perform a task when all lethality risks are removed from the system.

For this analysis, we used the case study colony whose genotype and phenotype are depicted in Figure 8. The results of the three measurements are shown in Table 2. All three measurements provide similar results for the age of the non-risky task (NAND). The results vary for the risky task. Specifically, the mean first age for task NOT (964) is substantially less than the mean age (1103.78), which, in turn, is substantially less than the mean age without lethality (1515.02). However, all three measurements capture the temporal polyethism structure in which task NAND is performed much earlier than task NOT within an organism's lifetime.

Discussion

In this paper, we have described how we have used Avida to explore a set of evolutionary conditions that give rise to temporal polyethism, a division of labor pattern. Specifically, we found that assigning different lethality risks to various types of tasks was a sufficient pressure to produce a temporal polyethism pattern, where organisms performed the least risky task earlier in their lifetime and then switched

Measurement	NOT	NAND
Mean Age	1103.78±25.93	236.43±5.69
Mean First Age	964±0	232.90±4.28
Mean Age No Lethality	1515.02±58.71	215.89±9.02

Table 2: Three different measurements of the age at which organisms perform a task. While all three have similar results for the non-risky task (NAND), the results differ a bit more for the risky task (NOT). However, all three measurements report a highly significant and substantial difference in mean ages between the two tasks and thus capture the temporal polyethism structure.

to performing the more risky task at the end of their life. This strategy balances a colony's need to maintain members of the colony and also to complete risky tasks. As such, this temporal polyethism structure enables the colony to be more efficient at gathering resources by having older organisms complete riskier tasks when they are closer to dying. In our analyses, we found further evidence that organisms made use of control flow instructions and genomic architecture modifications to achieve this behavior.

While our study sheds light on the evolutionary pressures that can give rise to a temporal polyethism pattern, the proximate mechanisms employed by colonies to exhibit this pattern could rely on either spatial structure (as proposed by the foraging for work hypothesis) or developmental hormones regulated by aging (as proposed by the task-riskiness hypothesis). For example, since the spatial structure of the nest corresponds with the riskiness of tasks, organisms may employ a foraging for work mechanism to achieve this pattern. Thus, workers may start within the nest taking care of the brood and then progress outward to more risky tasks, such as guard, undertaker, or forager (Hölldobler and Wilson, 2009). Even within Tofts' foraging for work model, workers switch between tasks based on colony need, and riskier tasks on the outside of the nest are a constant draw for work, trapping older workers outside of the nest (Tofts, 1993; Robson and Beshers, 1997).

Task switching may also be regulated by age using a variety of developmental hormones. Juvenile hormone (JH) is considered a mediator for temporal polyethism in advanced eusocial insects and even in some primitive wasps (Robinson, 1987; Shorter and Tibbetts, 2009; Sullivan et al., 2000). Studies of honeybees and some species of wasps show that when workers were treated with JH, they transitioned from nursing to foraging earlier in life (Robinson, 1987; Shorter and Tibbetts, 2009; Sullivan et al., 2000). In particular, honeybees have higher concentration of JH when they are older and foraging than they do when they are younger and taking care of the brood (Shorter and Tibbetts, 2009). Knocking down vitellogenin, a gene associated with JH, in bees similarly results in earlier task switching to foraging and shorter lifespans (Nelson et al., 2007). This example highlights

how developmental genes can regulate the performance of risky tasks so that they are done later in life and increase worker bee longevity. This proximate mechanism is compatible with the evolutionary pressures associated with ordering tasks according to risk.

An additional pressure that may reinforce ordering the performance of tasks according to risk is the benefit of conserving viable reproductives within the colony. In species in which workers have the option of reproducing when the queen dies, younger workers may have viable eggs and higher reproductive success than older sisters. By having younger workers perform safer tasks within the nest, the colony as a whole preserves its reproductive potential (Sendova-Franks and Franks, 1993).

Within this study, we have demonstrated that associating tasks with lethality risks is sufficient for evolving a temporal polyethism pattern. In the future, we will explore the effect of adding additional tasks and levels of risk. In addition, we will add in task-switching costs to address a limitation of Tofts' model, which assumes (unrealistically) that workers can switch between tasks without any delays. The evolutionary conditions leading to the rise of temporal polyethism is an important step in understanding the division of labor patterns we see in eusocial insects.

Acknowledgements

This work has been supported in part by NSF grants CCF-0643952, DBI-0939454, CCF-0820220, OCI-1122620, DGE-0718124.

References

- Crespi, B. J. (2001). The evolution of social behavior in microorganisms. *Trends in Ecology & Evolution*, 16(4):178–183.
- Duffy, J. E. (2003). The ecology and evolution of eusociality in sponge-dwelling shrimp. In *Genes, Behaviors and Evolution of Social Insects*. Hokkaido University Press.
- Franks, N. R., Tofts, C., and Sendova-Franks, A. B. (1997). Studies of the division of labour: neither physics nor stamp collecting. *Animal Behaviour*, 53(1):219–224.
- Goldsby, H. J., Dornhaus, A., Kerr, B., and Ofria, C. (2012). Task-switching costs promote the evolution of division of labor. In *preparation*.
- Hölldobler, B. and Wilson, E. O. (2009). *The superorganism: the beauty, elegance, and strangeness of insect societies*. WW Norton & Company.
- Jandt, J. M. and Dornhaus, A. (2009). Spatial organization and division of labour in the bumblebee *bombus impatiens*. *Animal Behaviour*, 77:641–651.
- Knoester, D. B., McKinley, P. K., Beckmann, B., and Ofria, C. (2007). Directed evolution of communication and cooperation in digital organisms. In *Proceedings of the European Conference on Artificial Life (ECAL)*.
- Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423:139–144.
- Nelson, C., Ihle, K., Fondrk, M., Page, R., and Amdam, G. (2007). The gene vitellogenin has multiple coordinating effects on social organization. *PLoS biology*, 5(3):e62.
- Ofria, C. and Wilke, C. O. (2004). Avida: A software platform for research in computational evolutionary biology. *Journal of Artificial Life*, 10:191–229.
- Pike, N. and Foster, W. A. (2008). *Ecology of Social Evolution*. Springer Berlin Heidelberg.
- Queller, D. C. and Strassmann, J. E. (2003). Eusociality. *Current Biology*, 13(22):R861–863.
- Robinson, G. (1987). Regulation of honey-bee age polyethism by juvenile-hormone. *Behavioral Ecology and Sociobiology*, 20(5):329–338.
- Robson, S. K. and Beshers, S. N. (1997). Division of labour and 'foraging for work': simulating reality versus the reality of simulations. *Animal Behaviour*, 53(1):214–218.
- Seeley, T. D. (1982). Adaptive significance of the age polyethism schedule in honeybee colonies. *Behavioral Ecology and Sociobiology*, 11(4):287–293.
- Sendova-Franks, A. and Franks, N. R. (1993). Task allocation in ant colonies within variable environments (a study of temporal polyethism: experimental). *Bulletin of Mathematical Biology*, 55(1):75–96.
- Shorter, J. and Tibbetts, E. (2009). The effect of juvenile hormone on temporal polyethism in the paper wasp *polistes dominulus*. *Insectes sociaux*, 56(1):7–13.
- Sullivan, J. P., Jassim, O., Fahrbach, S. E., and Robinson, G. E. (2000). Juvenile hormone paces behavioral development in the adult worker honey bee. *Hormones and Behavior*, 37(1):1–14.
- Tofilski, A. (2002). Influence of age polyethism on longevity of workers in social insects. *Behavioral Ecology and Sociobiology*, 51:234–237. 10.1007/s00265-001-0429-z.
- Tofts, C. (1993). Algorithms for task allocation in ants.(A study of temporal polyethism: theory). *Bulletin of Mathematical Biology*, 55(5):891–918.
- Traniello, J. F. and Rosengaus, R. B. (1997). Ecology, evolution and division of labour in social insects. *Animal Behaviour*, 53(1):209–213.
- Wilson, E. O. (1980). Caste and Division of Labor in Leaf-Cutter Ants (Hymenoptera: Formicidae: Atta): I. The Overall Pattern in *A. sexdens*. *Behavioral Ecology and Sociobiology*, 7(2):143–156.

Polarization and Belief Dynamics in the Black and White Communities: An Agent-Based Network Model from the Data

Patrick Grim,^{1,3} Stephen B. Thomas,^{2,6} Steven Fisher,³ Christopher Reade,⁴ Daniel J. Singer,⁵ Mary A. Garza,^{2,7} Craig S. Fryer^{2,7} and Jamie Chatman⁸

¹Group for Logic & Formal Semantics, Philosophy, Stony Brook University

²Maryland Center for Health Equity, School of Public Health, Univ. of Maryland

³Center for Study of Complex Systems, University of Michigan

⁴Gerald R. Ford School of Public Policy, University of Michigan

⁵Department of Philosophy, University of Michigan

⁶Department of Health Services Administration, School of Public Health, University of Maryland

⁷Department of Behavioral and Community Health, School of Public Health, University of Maryland

⁸Department of Biostatistics, Graduate School of Public Health, University of Pittsburgh

pgrim@notes.cc.sunysb.edu

Abstract

Public health care interventions—regarding vaccination, obesity, and HIV, for example—standardly take the form of information dissemination across a community. But information networks can vary importantly between different ethnic communities, as can levels of trust in information from different sources. We use data from the Greater Pittsburgh Random Household Health Survey to construct models of information networks for White and Black communities—models which reflect the degree of information contact between individuals, with degrees of trust in information from various sources correlated with positions in that social network. With simple assumptions regarding belief change and social reinforcement, we use those modeled networks to build dynamic agent-based models of how information can be expected to flow and how beliefs can be expected to change across each community. With contrasting information from governmental and religious sources, the results show importantly different dynamic patterns of belief polarization within the two communities.

Introduction

Does information move differently in the Black community compared to the White community? What kinds of informational contacts link family and friends in the Black community? What are the levels of trust regarding information from personal contacts, from the government, and from church or religious leaders? What is the information network characteristic of the two communities, and what are the levels of trust in various information sources? Given different informational input to those networks, what can we expect the dynamics of belief formation and change to be in the two communities?

We use data from the Greater Pittsburgh Random Household Health Survey (Sellars, Garza, Fryer & Thomas 2010) in order to construct models of information networks for White and Black communities, with levels of trust in various information sources correlated to position in those

networks. With simple assumptions regarding belief change and social reinforcement, we use those social networks to build dynamic agent-based models of how information flows and beliefs change across each community. These modeling results, abstract in character and yet grounded in data, show that contrasting information from governmental and religious sources can be expected to produce importantly different configurations of belief and belief polarization within the two communities.

What we are after in the long term is an understanding of how public health interventions might utilize belief dynamics to optimize information flow across existing social networks. More specifically, the aim is to focus attention on the role of trust and distrust that drives the persistent problem of racial and ethnic disparities in health and health care (Smedley, Stith, & Nelson 2003; Egede & Zheng 2003; Chen, Fox, Cantrell, Stockdale, & Kagawa-Singer 2007; Thomas & Quinn 2008; Corbie-Smith, Thomas & St. George 2002; Musa, Schulz, Harris, Silverman & Thomas 2009; Rajakumar, Thomas, Musa, Almario, & Garza 2009).

Information Networks from the Data:

Methods and Results

The Greater Pittsburgh Random Household Health Survey was conducted for the University of Pittsburgh Research Center of Excellence on Minority Health and Health Disparities via telephone by International Communications Research (ICR), an independent research company. Interviews were conducted with 1018 respondents age 18 or older. Of those respondents, 671 self-identified as African American/Black and 347 as Caucasian/White.

The survey was a large one, with questions regarding self-esteem, social support, trust, experiences of discrimination, religious involvement, depression, violence, physical activity, and health issues. It was not originally designed for purposes of either network analysis or agent-based modeling, but there

were several questions that allowed us to draw statistical data appropriate for these analyses. Among the questions asked of both Blacks and Whites, were two regarding social contact and support (Lubben, Gironde & Lee 2001):

1. How many relatives do you feel at ease with that you can talk about private matters? Would you say:

None? One? Two? Three or four? Five through eight? Nine or more?

2. How many friends do you feel at ease with that you can talk about private matters? Would you say:

None? One? Two? Three or four? Five through eight? Nine or more?

We combined answers for the two questions, giving a self-estimated total for each individual of how many friends or family they felt at ease with talking about private matters. Do African Americans report a wider or narrower net of contacts than Whites? A different distribution of contact types? We developed an algorithm to give us arbitrary networks of 100 nodes which instantiated the degree distributions evident in the data. We have since found several other effective algorithms in the literature (Badham and Stocker 2010). An animation showing progressive approximation to a set of degree distributions from the data set can be seen at www.pgrim.org/belief_dynamics.

Figure 1 shows a histogram of degree distributions for the Black and White community. The top row of boxes represents the degree distributions drawn directly from the survey data. The bottom row of boxes shows the approximations to those distributions we are able to achieve in construction of our artificial networks. Figure 1 also shows the artificial networks themselves, with nodes ordered from center to periphery in terms of number of connections.

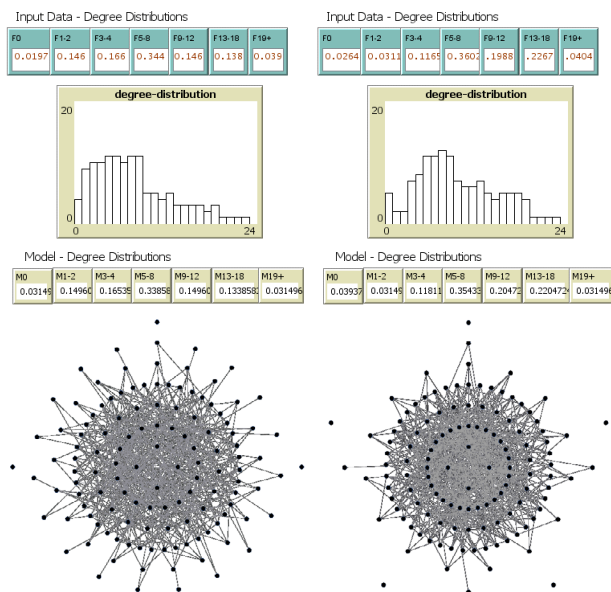


Figure 1 Friends and family networks in the Black (left) and White (right) communities.

Although a larger percentage of Whites report no family or friend contacts than Blacks do, a smaller percentage of Whites report only one or two friend or family contacts. The Black histogram offers a smoother curve, but shows a lower number of reported family and friend contacts over all. From the network diagrams it is evident that the Black information network is less tightly drawn: more nodes have fewer connections, and there are fewer numbers of nodes with large numbers of connections. Over all, the Black information network with family and friends appears to be sparser and more diffuse than that of the White community.

Although we have data on how many contacts each of our respondents reported, and although our model constructs a network that matches those numbers, our current data does not offer any information about other aspects of network structure—correlation coefficient, for example.

Trust

Our modeled networks reflect different patterns of information contact between friends and family within the contacts between friends and family within the White and Black communities. Information from those contacts can be expected to have a major impact on belief formation, but individuals also get information from other sources. The influence of information from any of these sources can be expected to vary with an individual's trust in the source.

One set of questions in the Greater Pittsburgh Random Household Health Survey targets trust. Among other sources, respondents were asked about their trust in information from the CDC, friends or family, and church or religious leaders:

3. There are many people, or groups, from whom you might get information about health or health problems. For each of the following, please indicate how much you, personally, feel you would trust information that you got from that source.

How about the Center for Disease Control, sometimes referred to as the CDC? Would you say you:

Would trust definitely?
Would trust probably?
Would not trust probably?
Would not trust definitely?

Response options were the same for:

How about your friends or family?
How about your church or religious leaders?

For the sake of simplicity, we grouped 'would trust definitely' and 'would trust probably' as a positive trust category and 'would not trust probably' and 'would not trust definitely' as a negative trust category.

The initial presentation of data from the Greater Pittsburgh Random Household Health Survey gave trust levels across the full aggregate. Our agent-based model is more finely tuned than that. It's not as if there are two isolated facts:

(a) that some individuals in each community have a wider contact net of family and friends, and

(b) that some individuals are more trusting of health information from particular sources.

There are correlations between these, evident in the raw data if not its initial presentation. We dug out those correlations in building the agent-based network model.

For those agents who reported no friend or family contacts, we incorporate appropriate percentages with positive or negative trust in information from various sources. For those reporting one or two contacts, we incorporate the different trust percentages appropriate to these, and so on.

Within the Black community, network diagrams of trust in information from (a) friends and family, (b) governmental sources, and (c) church or religious leaders are shown in Figure 2. Corresponding trust levels for each source of information for the White community are shown in Figure 3.

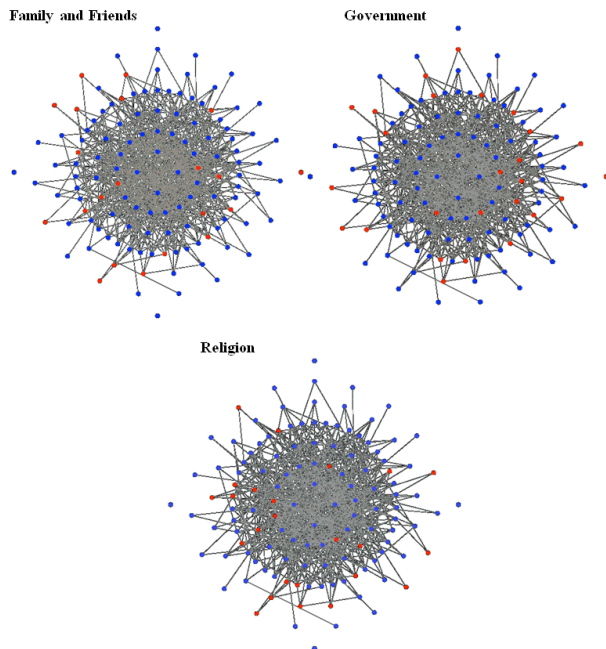


Fig. 2 Trust levels in information sources in the Black community. Blue = positive trust. Red = Distrust

Distrust of family and friends is tied more clearly to isolation from family and friends in the Black than in the White community. Many of those with only one or two contacts report distrust of family and friends within the Black community, whereas none of those with only one or two contacts do so in the White community. Distrust of government is more widespread within the Black community and is evident across most levels of connection.

Most noticeable, however, are differences in trust of church and religious leaders. Distrust of these information sources is much higher in the White community than in the Black community. Distrust of religious sources is also more strongly represented among those with many informational connections among the White compared to the Black community.

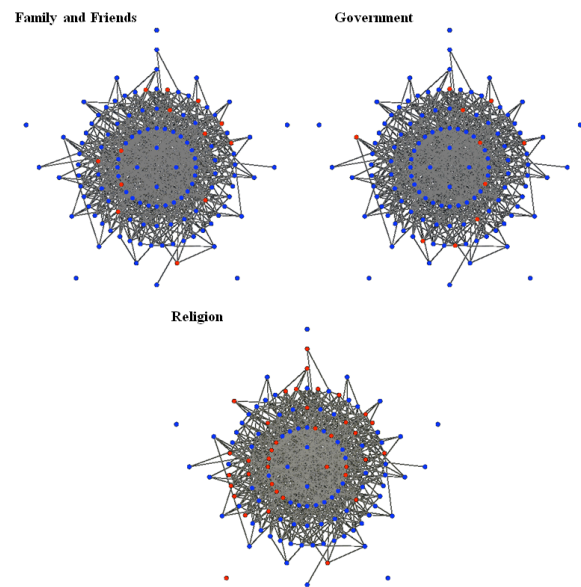


Fig. 3 Trust levels in information sources in the White community. Blue = positive trust. Red = Distrust

Information Dynamics: Methods

The networks constructed above, with correlated trust levels, allow us to project a dynamic model of belief across the two communities. Our aim is to offer an abstract model of how, given different information structures and different trust levels, the same information from external sources may result in different dynamics and different eventual configurations of community belief. How, for example, might conflicting health information from governmental and religious sources impact the dynamics and polarization of health care beliefs within the Black and White communities?

The data from which we have built the network model above is a snapshot of attitudes at a particular point in time. From that we can go on to construct a dynamic model, capable of offering a projection of potential changes in attitude over time. The fact that dynamic modeling can build on but also take us beyond static data carries pitfalls as well as promise. In order to test dynamic projections of the model in full we would need longitudinal data on changes in attitudes toward a particular health measure in the two communities correlated with data on information sources over the period at issue. That is longitudinal data we do not have and that we are unlikely to be able to get. In the absence of full longitudinal validation, we need to be particularly sensitive to the assumptions that drive dynamic projections.

A primary assumption in the construction of our dynamic model here is a mechanism for belief updating. We begin with the networks outlined above for each community: networks of contact which match degree distributions drawn from the data, correlated with trust levels regarding information from (a) friends and family, (b) governmental sources, and (c) church and religious leaders. What we want to know is how the structure of the information network and

inputs from these sources affect the belief configuration of the community over time.

We model partial or gradational beliefs with numbers between 0 and 1. These might represent the agents' degrees of confidence that they will catch a disease, for example, or their estimates of the severity of a disease (Harrison, Mullen, & Green 1992; Janz & Becker 1984; Mullen, Hersey, and Iverson, 1987; Strecher & Rosenstock 1997). At the high end, these numbers might represent a belief that infection is imminent (represented by 1), which thereby warrants vaccination; at the low end, they might represent a belief that infection is impossible (represented by 0), and so vaccination is unwarranted; in such a case .5 might represent a neutral degree in between. Nothing in the model, however, indicates what types of belief are at issue or how the numerical scale is to be read. We abstract from the particular beliefs at issue, using numbers in their stead.

Agents update their beliefs, in our model, in light of information from family and friends, governmental sources, and church and religious leaders. How much they are influenced by each source will depend on how much trust they put in each source. At each step in the dynamic development of the model each agent considers input from (a) friends and family, weighted by how much trust he or she has in friends and family, input from (b) governmental sources, weighted by how much trust he or she has in government, and (c) from church and religious sources, again weighted by trust. These minimal assumptions, we can argue, are at least relatively realistic: people do have beliefs some of which can be represented on such a scale, and people are influenced to change those beliefs by, among other things, the expressed beliefs of those with whom they have contact and information that they trust from external sources. Given the networks of information contacts modeled above, it is clear that there will be reinforcement effects in such a dynamic. The fact that two trusting friends converge on a belief will strengthen that belief in both, for example. The fact that most of one's friends hold a belief will have a stronger effect than if only one does.

Our model starts, therefore, with a randomized distribution of beliefs. At each successive step, agents will have shifted their beliefs. They will then have different input from family and friends (though input from governmental and religious sources remain the same), producing a further modification of beliefs. What we track the change of belief and belief polarization over time in the two communities.

Although the general patterns of contact reinforcement and influence from outside forces can be seen as minimal and plausible assumptions of the model, the specific way in which these are instantiated in belief updating must be seen explicitly as modeling abstractions and simplifications. Our model is built on simplified assumptions regarding (1) the relative balance of various information sources and (2) the treatment of survey information on distrust. In this model, we use a simple weighted average in order to balance different information sources. Our basic updating algorithm is one in which current belief carries the largest weight in influencing later belief. Input from friends and family as a whole count half that weight in updating, with information from governmental and religious sources each counting one quarter. At each iteration, our agents average their current belief with input from each of these sources weighted in these

proportions, resulting in their belief at the next iteration. That basic algorithm is altered slightly so as to indicate greater influence from greater numbers of contacts: for each of 5 categories of multiple friends (3-4, 5-8, 9-12, 13-18, and > 18) the influence of friends and family is increased by 10% over the base rate. The algorithm is also significantly altered by trust levels. In this model we simply discount sources an individual 'distrusts': governmental input to an individual who distrusts the government, for example, is simply ignored. In further studies we also explore interpreting reported distrust as a negative weighting for information from a particular source.

The updating algorithm we use, in the tradition of French 1956, Harary 1959, DeGroot 1974, and Golub & Jackson 2010, 2011 and forthcoming, and is compatible with many standard accounts of partial belief dynamics including Bayesian conditionalization. In the most natural scheme for thinking of our agents' beliefs in Bayesian terms, there may be an expectation at the extremes, but see Hájek 2003. The use of weighted averaging in the updating algorithm could also be seen as a natural extension of the popular Equal Weight View in the literature on peer disagreement (Feldman 2006, Elga 2007, and Christenson 2007).

No-one thinks that weighted averaging of beliefs in an informational neighborhood—let alone these specific weights—captures the full psychological or normative dynamics of belief. Such a mechanism is a modeling abstraction intended to capture patterns of reinforcement which in some form clearly are plausible aspects of belief change. The more trusted an information source, the more likely information from that source is to change one's beliefs. The more one's beliefs are like those of one's network neighbors, and the more they are like more of one's network neighbors, the less inclination there will be to change those beliefs. The more one's beliefs are out of sync with one's neighbors, the greater the pressure there will be to change one's beliefs. That beliefs will change in accord with outside information and some pattern of reinforcement along those lines is very plausible, backed by a range of social psychological data, and is therefore an aspect of realism in the model. What is purely an assumption of the model is the particular algorithm used for reinforcement and informational influence—the particularly simple pattern of weighted belief averaging, applied homogeneously across agents.

In order to be informative regarding an exterior reality, a model, like any theory, must capture relevant aspects of that reality. In order to offer both tractability and understanding, a model, like any theory, must simplify. Our attempt is to capture some predictable but general aspects of belief change and reinforcement across a community; the admittedly artificial assumption of the specific algorithm we've used for belief updating is our simplification.

Information Dynamics and Polarization in the Black and White Communities

What can be projected for the Black community with belief change on this model and networks structure and trust levels derived from our data? How do beliefs change over time with particular governmental and religious inputs?

Figure 4 shows the modeled development of beliefs across the Black community in terms of a histogram of the number of agents holding a belief in a particular category over time. In this case we use an input of '1' from governmental sources, '0' from religious sources, reflecting development in a case in which health care information from church and religious sources was directly opposed to that from the government. A full animation of such a development, correlated with node changes in the network, is available at www.pgrim.org/belief_dynamics.

The resultant belief configuration in this simulation has a mean of .48—slightly less than the mean of the random beliefs with which we began. It is the distribution of those beliefs that is particularly interesting, however. The result shows a clear central consensus, but development of the model shows increasing polarity, resulting in an obvious polarization at the two ends. If governmental sources say one thing and religious sources say another, our model indicates that the Black community will have a central consensus but a significant number of people with beliefs polarized at the extreme ends.

How does this development compare with the same inputs for the White community? Histograms of belief distribution over time for the White community are shown in Figure 5.

In this case the final mean for the community is .62 as opposed to .48. The model projection in a case of polarized information from governmental and religious sources is that governmental information will trump religious sources in the White community: belief in the White community will tend significantly toward that promulgated by the government. Within the Black community, in contrast, the two influences will be roughly on a par.

Here again, however, it is the distribution of beliefs that is equally or more important. In the White community the central consensus is significantly less sharp. In almost all runs it carries a secondary bump to the right of a central consensus, as shown here. Polarization at both extremes is significantly less in the White community: it is only the governmental end that shows a pile-up comparable to both ends in the Black community.

Our model therefore projects important differences in dynamics and final configuration of beliefs within the Black and White communities given the same polarized input from religious and governmental sources. Central consensus is more unified in the Black community, though with a more significant percentage of the population fully polarized and roughly equally balanced at the religious or governmental ends. The White community shows a less centralized consensus. Both in central areas and in polarized ends, it is governmental information that has a greater effect within the White community.

The results above use an input of '1' for governmental and '0' for religious informational sources. If both religious and governmental inputs are '1', progressive weighted averaging of inputs will drive consensus entirely to the '1' side. If both inputs are '0', that mechanism will drive consensus to the '0' side. The interesting results are therefore those in which we have differences in the two inputs. Our model can be run for any values of these, however, and need not be 'all or nothing'. With an input of .33 from one side and .66 on the other, a

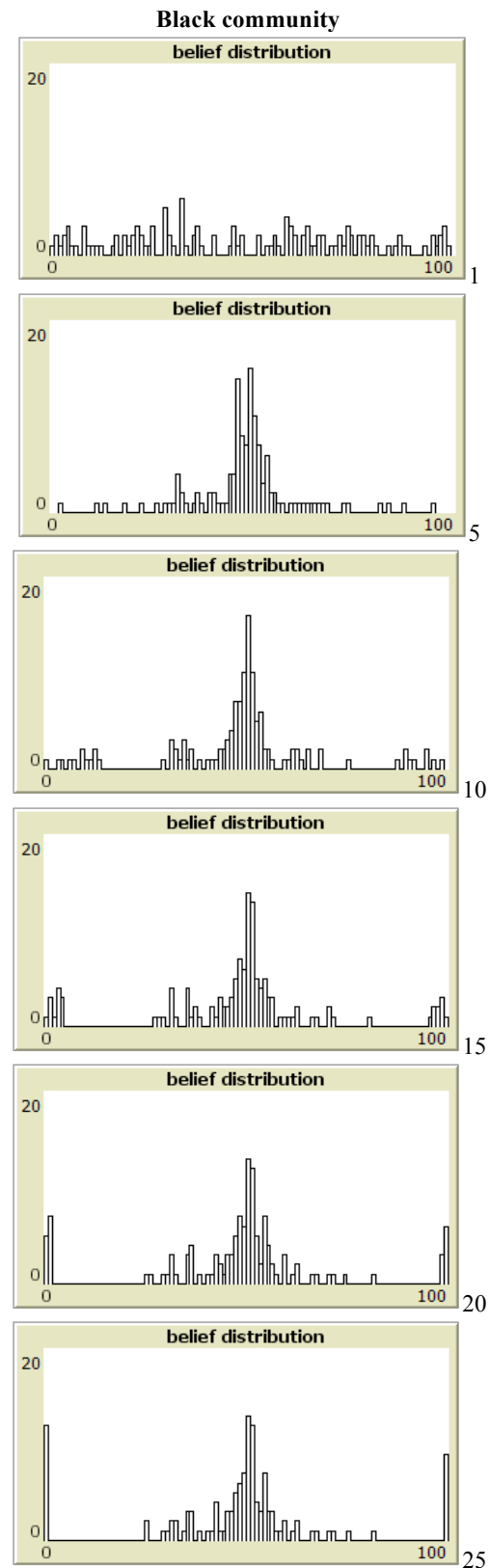


Fig. 4 Black community: dynamics of belief distribution given governmental input = 1, religious input = 0, iterations as numbered.

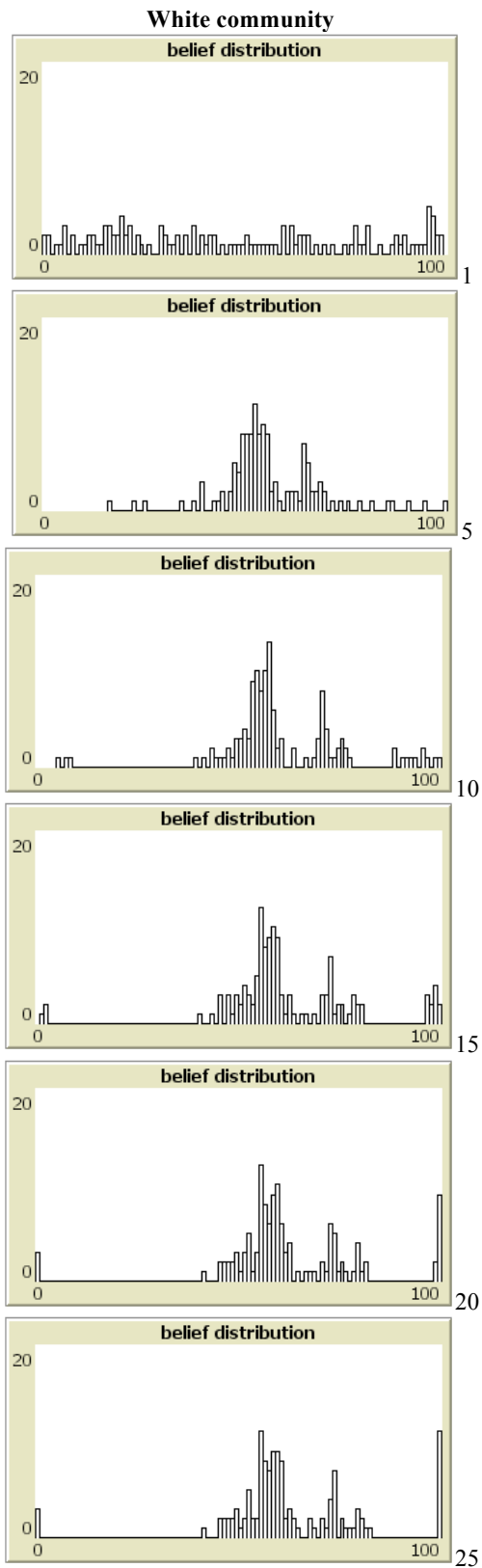


Fig. 5 White community: dynamics of belief distribution given governmental input = 1, religious input = 0, iterations as numbered.

similar pattern of polarization is evident, but with the poles at .33 and .66 rather than at 0 and 1.

In Figure 6 we take comparison of runs for different inputs one step further. For each combination of inputs from religious and governmental sources, at .1 intervals, we ran 100 simulations and took medians and quartiles across those runs. The top landscape in Figure 6 shows the pattern of medians for the Black community across different inputs from governmental and religious sources. The lower landscape shows the corresponding pattern for the White community with that range of inputs. Together the two show the important tilt of the White community toward input from governmental sources when compared with the Black community.

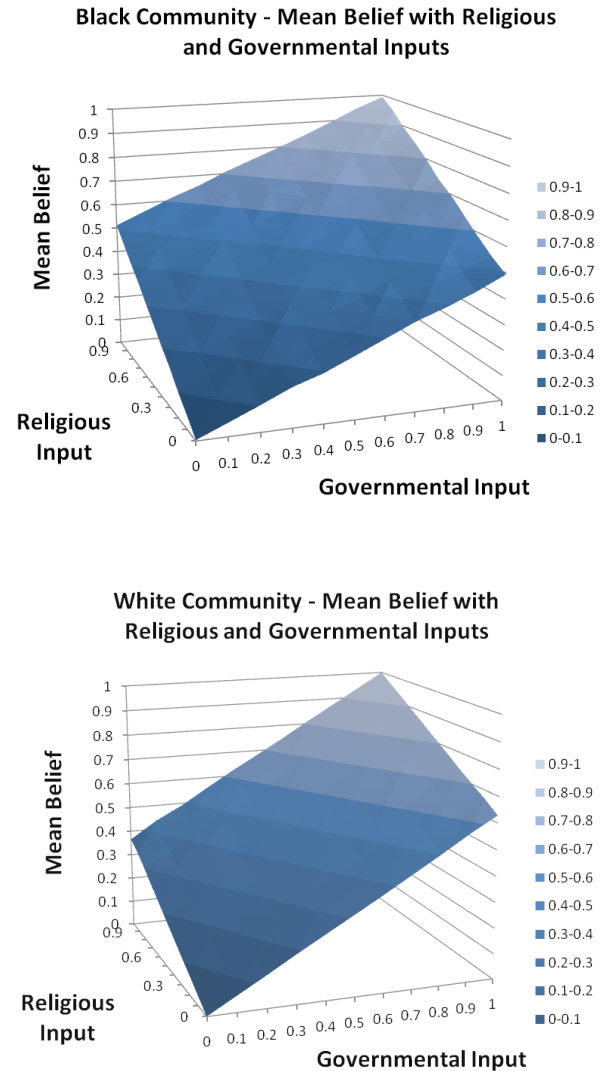


Fig. 6 Differences in median belief between the two communities across a range of different inputs from governmental and religious information sources.

A slice through the (1,1) diagonal on each of these is shown in Figure 7, here including the 25th and 75th quartile for each

population. This again makes vivid the differences in polarization between the communities at the same points of input from religious and governmental sources.

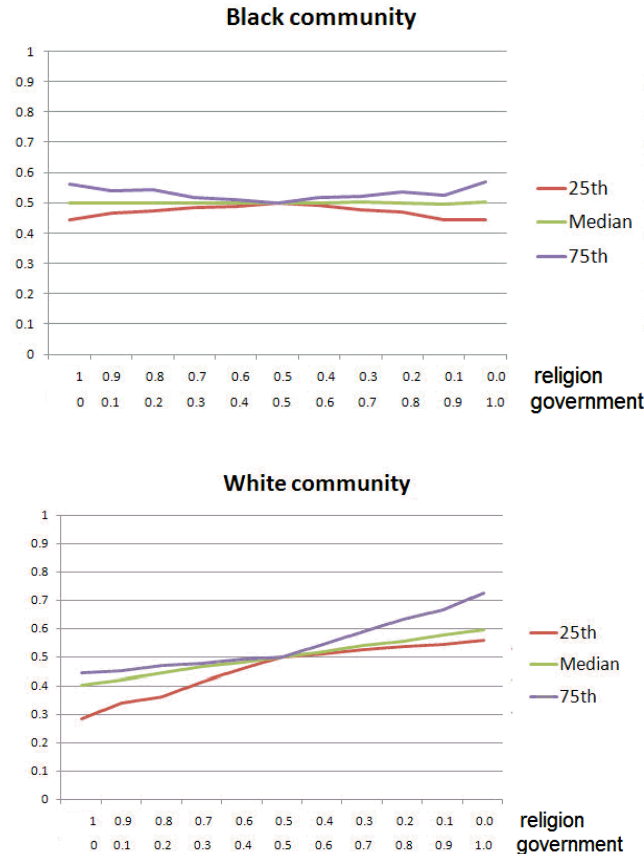


Fig. 7 Medians and quartiles for White and black Communities with different combinations of input from governmental and religious information sources.

Conclusion

Dynamic agent-based modeling, constructed on social networks of interaction drawn from the actual data, demonstrates important divergences in social reaction to particular patterns of information within the Black and White communities. Surprise has often been expressed that Black and White communities have reacted differently given the same exterior information, particularly from governmental and or religious sources. The portrait of different social information structures offered here, incorporating network contact patterns that can differently amplify differences in trust, should reduce that element of surprise. This form of analysis can both offer a projection of differences in belief dynamics in future cases and might be used to best target effective information interventions in public health.

Our target is an understanding of the social dynamics of belief, a target we think clearly belongs under the wide umbrella of social epistemology. Because we want to understand the real social factors in belief formation, we've

based our study in real data. In order to project longitudinal patterns from a static data snapshot, however, and in order to explore 'what ifs' relevant to normative questions of intervention, we've employed a range of simplifying assumptions within the techniques of dynamic agent-based modeling.

Passage of the historic Health Care and Education Reconciliation Act of 2010 (H.R. 4872) (US Congress 2010) and launch of Healthy People 2020 (US Dept. of Health & Human Services 2009) provides an opportunity for multiple disciplines to collaborate on solutions to eliminate racial and ethnic health disparities. We believe this hybrid of disciplines and techniques can serve as an example for further research: work both data-driven and model-instantiated, both descriptive and normative, putting abstract techniques to the practical mission of eliminating health disparities and achieving health equity for all.

Acknowledgements

This work was supported in part by the National Institute of General Medical Sciences MIDAS (1U54GM088491-01) Computational Models of Infectious Disease Threats, and by a pilot grant for Developing an Agent-Based model to Assess Racial Differences in Medical Discrimination, Social Support and Trust, administered by the Graduate School for Public Health at the University of Pittsburgh. Work was also supported in part by the National Institute on Minority Health and Health Disparities (PG60MD000207, S. B. Thomas, PI) and the National Cancer Institute (K01CA148789, C. S. Fryer, PI and K01CA140358, M. A. Garza, PI). Grim is grateful for time at the Center for Complex Systems and the Department of Philosophy at the University of Michigan. Dr. Chatman's participation was supported in part by the Kellogg Health Disparity Scholars Program.

References

- Badham, J., and Stocker, R. (2010). A spatial approach to network generation for three properties: Degree distribution, clustering coefficient and degree assortativity. *Journal of Artificial Societies and Social Simulation* 13, <http://jasss.soc.ac.uk/13/1/11.html>.
- Chen, J., Fox, S., Cantrell, C., Stockdale, S., and Kagawa-Singer, M. (2007). Health Disparities and prevention: racial/ethnic barriers to flu vaccinations. *Journal of Community Health* 32, 5-20.
- Christensen, D. (2007). Epistemology of disagreement: the good news. *Philosophical Review* 119: 187-217.
- Corbie-Smith, G., Thomas, S. B., St. George, S. M. M. (2002). Distrust, race, and research. *Archives of Internal Medicine* 162: 2458-2463.
- DeGroot, M. H. (1974). Reaching a consensus. *Journal of the American Statistical Association* 69, 118-121.
- Egede, L. and Zhen, D. (2003). Racial/ethnic differences in influenza vaccination coverage in high-risk adults. *American Journal of Public Health* 93, 2074-2078.
- Elga, A. (2007). Reflection and disagreement. *Noûs* 41: 478-502.
- Feldman, R. (2006). Epistemological puzzles about disagreement. In Hetherington, S., editor, *Epistemology Futures*. Oxford: Oxford University Press, pp. 216-236.
- French, J. (1956). A formal theory of social power. *Psychological Review* 63, 181-194.
- Golub, B., and Jackson, M.O. (2010). Naive learning in social networks: Convergence, Influence, and the Wisdom of Crowds. *American Economic Journal: Microeconomics* 2, 112-149.

- Golub, B., and Jackson, M. O. (forthcoming a). Network structure and the speed of leaning: Measuring homophily based on its consequences. *Annals of Economics and Statistics*. Available at <http://papers.ssrn.com/abstract=1784542>.
- Golub, B., and Jackson, M. O. (forthcoming b). How homophily affects the speed of learning and best response dynamics. *Quarterly Journal of Economics*.
- Harrison, J. A., Mullen, P. D., and Green, L. W. (1992). A meta-analysis of studies of the health belief model. *Health Education Research* 7: 107-116.
- Hájek, A. (2003). What conditional probability could not be. *Synthese* 137 (3): 273-323.
- Harary, F. (1959). A criterion for unanimity in French's theory of social power. In Cartwright, D., editor, *Studies in Social Power*. Ann Arbor: University of Michigan Press.
- Lubben, J. E., Gironde, M., and Lee, A. (2001). Refinements to the Lubben Social Network Scale: The LSNS-R. *Behavioral Measurement Letter* 7: 2-11.
- Mullen, P. D., Hersey, J., and Iverson, D. C. (1987). Health behavior models compared. *Social Science and Medicine* 24: 973-981.
- Musa, D., Schulz, R., Harris, R., Silverman, M., and Thomas, S. B. (2009). Trust in the health care system and the use of preventive health services by older black and white adults. *American Journal of Public Health* 99 (7): 1293-1299.
- Rajakumar, K., Thomas, S. B., Musa, D., Almario, D., and Garza, M. (2009). Racial Differences in parents' distrust of medicine and research. *Archives of Pediatric and Adolescent Medicine* 163(2) 108-114.
- Sellars, B., Garza, M. A., Fryer, C. S. and Thomas, S. B. (2010). Utilization of health care services and willingness to participate in future medical research: The role of race and social support. *Journal of the National Medical Association* 102: 776-786.
- Smedley, B., Stith, A., and Nelson, A. (2003). Eds, *Unequal Treatment: Confronting Racial and Ethnic Disparities in Health Care*, Committee on Understanding and Eliminating Racial and Ethnic Disparities in Health Care, Institute of Medicine of the National Academies, Academies Press.
- Strecher, V., J. and Rosenstock, I. M. (1997). The health belief model. In Glanz, K., Lewis, F. M., and Rimer, B. K., *Health Behavior and Health Education: Theory, Research and Practice*, San Francisco: Jossey-Bass.
- Thomas, S., and Quinn, S. (2008). Poverty and elimination of urban health disparities: challenge and opportunity. *Annals of the New York Academy of Sciences* 1136: 111-125.

The Effects of Finite Populations and Selection on the Emergence of Signaling

Kyle I. Harrington, A. Pinar Ozisik, Jordan B. Pollack

DEMO Lab, Computer Science Department, Brandeis University, Waltham, MA 02454

Abstract

In the research described here we examine the emergence of signaling from non-communicative origins, using the Sir Philip Sidney Game as a framework for our analysis. This game is known to exhibit a number of interesting dynamics. In our study, we quantify the difficulty of reaching multiple types of equilibria from initially non-communicative populations with an infinite population model. We then compare the ability of finite populations with typical tournament selection to approximate the behaviors observed in infinite populations. Our findings suggest that honest signaling equilibria are difficult to reach from non-communicative origins. In the second part of the paper, we show that the finite model fails to model dynamics that permit deceptive signaling under typical evolutionary conditions, where infinite populations exhibit spiraling behavior between honest and deceptive signaling.

Introduction

Communication and expression in man and animals has allowed for the formation of complex social organizations. Although sophisticated forms of communication have emerged, such as human language, the origin of animal communication is rooted in the exchange of simple signals. These signals have coevolved between senders and receivers for the communication of attributes such as need, status, and intention. We study a simple signaling game that allows us to address some shortcomings of previous studies on the emergence of signaling. We quantify the difficulty of evolving honest signaling systems, and the failure of some finite models to permit deceptive dynamics. These observations are a step towards understanding how a rational agent could respond to a signal with Sir Philip Sidney's immortal words: "thy necessity is yet greater than mine."

The coevolution of signaling has attracted attention since the inception of the field of artificial life and even earlier in studies of animal behavior and ethology. Evolutionary computation researchers studying the origins of signaling generally employ evolutionary algorithms (EA) in their models, while game theorists employ population dynamics models and analytical tools. Some EA work has used game-theoretic analysis to constrain parameters (Bullock, 1997);

however, we are not aware of any studies of coevolved signals that relate continuous population dynamics to EA dynamics for signaling games. We investigate this relationship and focus on similar EA configurations to those used in previous studies of the emergence of signaling. In particular, we consider the discrepancy between the dynamics of finite population EA's and continuous population dynamics.

The relationship between continuous and finite population evolutionary dynamics has been a contentious topic (Fogel and Fogel, 1995; Ficici et al., 2005; Ficici, 2006; Ficici and Pollack, 2007; Nowak et al., 2004). An evolutionarily stable strategy (ESS) is defined for continuous population dynamics as a strategy that cannot be invaded by a rare mutant (Maynard Smith and Price, 1973; Maynard Smith, 1982). A common question in the study of evolutionary dynamics is when finite populations can achieve an ESS. In particular, the two discoveries that inspire this study are: Best-of-group tournament selection cannot converge to polymorphic Nash equilibria (Ficici et al., 2005), and even with a good selection method, a finite population may be too small to maintain an ESS (Ficici and Pollack, 2007). In a simple signaling game we investigate both the reachability of interesting equilibria from non-communicative origins, and compare the coevolution of continuous population dynamics to finite populations under tournament selection (the most common selection method used in previous work). We find that multiple dynamics involving signaling behavior are more easily reached than the traditional signaling ESS, and one of these dynamics is poorly represented by a finite population.

Background

Zahavi introduced the idea of costly signals as handicaps which lead to reliable signals (Zahavi, 1975). This handicap principle has been used to explain how signaling attributes which would seem to be energetically expensive or superfluous for survival can be selected for, especially in sexual selection. For example, plumage like the peacock's tail signal virility and strength to a peahen because the male has honestly demonstrated that it can carry the unneeded weight of the brilliant tail. For a good web exposition on honest

Table 1: The Sir Philip Sidney game.

(a) Payoff matrix.			(b) Sender strategies.		(c) Donor strategies.	
	Donate	Keep	ID	Signaler strategies	ID	Donor strategies
Potential donor	$1 - d$	1	<i>SH</i>	signal only if healthy	<i>DQ</i>	donate only if no signal
Signaler			<i>ST</i>	signal only if thirsty	<i>DS</i>	donate only if signal
Thirsty	1	$1 - a$	<i>SN</i>	never signal	<i>DN</i>	never donate
Healthy	1	$1 - b$	<i>SA</i>	always signal	<i>DA</i>	always donate
$m = p(\text{thirsty}) = 0.5$	Signal cost = c					

signaling, see (Bergstrom, 2012). This work was later given a rigorous mathematical treatment in (Grafen, 1990) for signals of a continuous range of quality. Later, two simple discrete signaling games were developed: the Sir Philip Sidney game (Maynard Smith, 1991) and the discrete action-response game (Hurd, 1995). The former can be seen as a generalization of the latter, which is a deliberately minimal signaling game. The discrete action-response game is based upon the handicap principle, thus models costly signaling. As the Sir Philip Sidney game is the subject of our study, we will introduce it in greater detail later.

Bullock analytically evaluated the discrete action-response game for parameters that should lead to the emergence of signaling, then used an EA to evolve a finite population (Bullock, 1997). Agents take turns playing an iterated signaling game, and are selected for reproduction using spatial tournaments. The results demonstrate a number of dynamics ranging from evolutionarily stable strategies (ESS) to cycles. It is found that the emergence of honest signaling from a non-communicative state only occurs from a subset of the analytically determined cooperative parameters.

Noble studied a version of the discrete action-response game where only one of the signaler states results in positive payoff for the sender and receiver (Noble, 1999). The criteria for the honest signaling ESS was shown to be when the payoff for signaling is greater than the cost of signaling, and the payoff of responding is greater than the cost of responding. Noble suggested that a signaling game must permit imperfect information, deception, and manipulation to allow for information transmission. While all of these points are present in the described game, we note that the ambivalence of signalers to transmit a signal in one of the two possible states means there is no incentive for deception. We demonstrate situations where an incentive to signal from both signaler states has significant implications on the coevolutionary dynamics of signaling.

We have previously worked on evolution of communication in a group foraging task, although without referencing the signaling literature (Saunders and Pollack, 1996). Similarly, Reggia et al. investigate conditions that enable the emergence of signaling (Reggia et al., 2001). In this work the authors use a 2D simulated world where agent behavior is governed by a finite-state machine, and signaling ability is

encoded in the genome. Agents have an energetic cost of living, and independent experiments are performed for predator signaling, food signaling, and environments where both types of signaling are possible. Their EA operates on a population of size 200 and multiple forms of tournament selection are compared. It is particularly interesting that smaller tournament sizes and spatially-constrained tournaments lead to more signaling. While the authors describe a set of conditions that enable signaling for their world/agent architecture, in this study we will investigate conditions that enable signaling in a simplified environment.

A review of the evolution of signaling systems is beyond the scope of this paper. For an extensive review of studies on simulating the emergence of communication, including signaling, see (Wagner et al., 2003).

Sir Philip Sidney Game

The Sir Philip Sidney (SPS) game was developed by John Maynard Smith as a model of costly signals (Maynard Smith, 1991). It is an extensive form game between two players. The importance of costly signals is based upon Zahavi's handicap principle (Zahavi, 1975) which states that reliable signals are costly with respect to the signaler's ecological context. This cost is explicitly introduced as a fitness penalty in the SPS game.

The SPS game is played for a single round between two players: a signaler and a donor. The signaler may be in one of two states: *thirsty* or *healthy*. The probability of the signaler being *thirsty* is m . A *thirsty* signaler has a fitness of $(1 - a)$, and a *healthy* signaler has a fitness of $(1 - b)$. In all cases $a > b$. The strategy of the signaler specifies whether it signals in either, both, or neither states. It costs the signaler c to transmit a signal. In response to receiving a signal the donor decides whether or not to donate to the signaler. Donation comes at a cost, d , to the donor, but heals the signaler to a fitness of 1. Furthermore, a globally-fixed relatedness term, r , is introduced which accounts for the opponent in the inclusive fitness of each player. Labels for signaler and donor strategies are listed in Tables 1(b) and 1(c), respectively. For example, in a game between *ST* and *DS*, if *thirsty* the signaler will transmit a signal and in response the donor donates. The signaler's fitness is $(1 - c + r(1 - d))$ and the donor's fitness is $(1 - d + r(1 - c))$. If the game

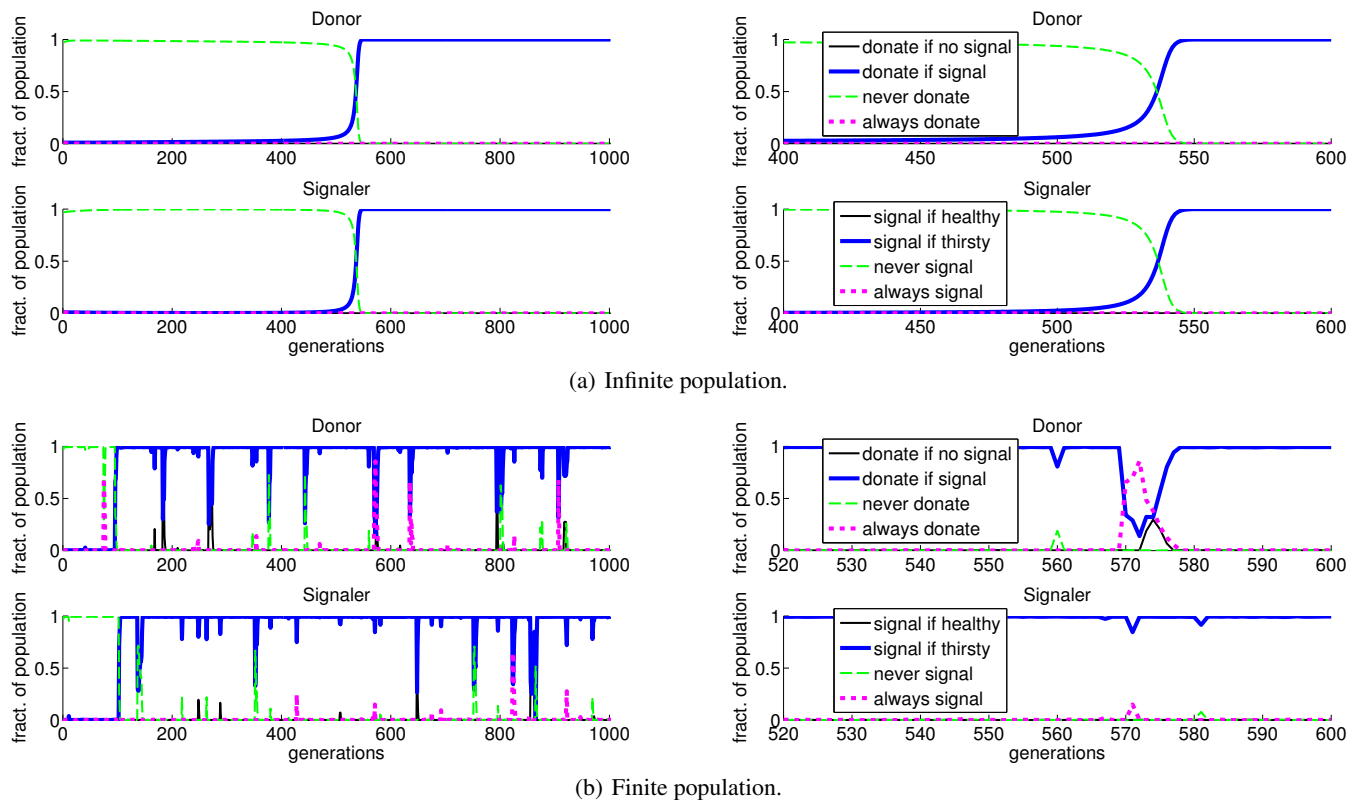


Figure 1: Example of an honest signaling equilibrium. The graphs on the right hand side are zoomed in versions of those on the left. In 1(b) the “always donate” strategy briefly invades the donor population. This is a possibility in the finite model if the SPS parameters are within a range where non-optimal strategies can be mistook for sampling noise.

was played between *ST* and *DQ*, then if *thirsty* the signaler transmits a signal and the donor does not donate. The signaler’s fitness is $(1 - a - c + r)$ and the donor’s fitness is $(1 + r(1 - a - c))$. The payoff matrix is shown in Table 1. Unless otherwise specified we set $m = 0.5$.

The SPS game has been the subject of a number of game theoretic studies, for both the discrete signaling game we study here, and the continuous-version of the SPS game (Johnstone and Grafen, 1992). The interest in this game arises from its facilities for modeling both costly signaling and signaling amongst relatives, where the latter property permits cost-free signaling in a number of conditions.

The key distinction between the discrete action-response (Hurd, 1995) and SPS games is the use of inclusive fitness (Hamilton, 1964), adding the opponent’s score weighted by a “relatedness” term, r . Relatedness accounts for the fact that if a player’s opponent is related to the player, then benefits to the opponent also benefit the player. Inclusive fitness is only utilized when computing the score for a donor and signaler playing a game, as opposed to fitness sharing from genetic algorithms where related individuals in the same population share the fitness of a given niche. In (Ozisk and Harrington, 2012) it was shown that relatedness based upon

tags, unique phenotypic identifiers, destabilizes honest signaling equilibria in finite models.

Non-communicative Equilibria

In this study we are interested in the emergence of signaling from non-communicative initial conditions. While there are multiple combinations of signaler and donor strategies that do not transfer information, we will be particularly interested in the *SN* and *DN* combination of strategies, because the two populations will be initially composed of predominately *SN* and *DN* individuals. Bergstrom and Lachmann (Bergstrom and Lachmann, 1997) have shown the *SN* and *DN* pair to be a Nash equilibrium if

$$d > r(ma + (1 - m)b)$$

Huttegger and Zollman (Huttegger and Zollman, 2010) note that reversing the inequality leads to the *SN* and *DA* pair of strategies being a Nash equilibrium. They refer to these as “pooling equilibria.”

Signaling Equilibria

One of most commonly studied type of equilibria in signaling games with handicap signals is the signaling ESS, sometimes referred to as separating equilibria. In these equilib-

ria the *ST* and *DS* strategies are dominant. Bergstrom and Lachmann (Bergstrom and Lachmann, 1997) show this is a Nash equilibrium when

$$a \geq c + rd \geq b \quad \text{and} \quad a \geq d/k \geq b.$$

An example of this type of signaling equilibrium is shown in Figure 1. We will refer to this type of signaling equilibrium as the honest signaling equilibrium.

Another type of signaling equilibrium is possible where the *SH* and *DQ* strategies are dominant. Huttegger and Zollman (Huttegger and Zollman, 2010) show this is a Nash equilibrium when

$$a \geq rd - c \geq b \quad \text{and} \quad a \geq d/k \geq b.$$

In previous work on evolving communicative agents we have seen this type of strategy pattern emerge (Saunders and Pollack, 1996). We will refer to this type of signaling equilibrium as the inverse honest signaling equilibrium.

Hybrid Equilibria

A dynamic of particular interest in the SPS game is that of hybrid equilibria, whose name is taken from the economics literature. First formally presented for the SPS game in (Huttegger and Zollman, 2010), hybrid equilibria are actually a family of polymorphic mixed Nash equilibria. In practice these hybrid equilibria can be observed in the SPS game as a spiraling phenomenon (Figure 2). The system first approaches a signaling equilibrium, such as *ST* and *DS*, and upon reaching a certain fraction of signalers and responsive donors *SA* signalers begin to take advantage of the donors. The introduction of these deceptive signalers into the population causes the *DN* strategy to increase in the donor population. As the *DN* strategy increases it becomes less favorable to signal. The *SA* strategy signals both when *thirsty* and *healthy*, as opposed to the *ST* strategy which only signals when *thirsty*, which means that the *SA* strategy has a lower fitness than *ST* when playing against the *DN* strategy, thus the *SA* strategy will be more strongly selected against. As *ST* begins to take over the signaler population the *DS* strategy also increases. Huttegger and Zollman (Huttegger and Zollman, 2010) show that the polymorphisms of the hybrid equilibria are mixed Nash equilibria given by $\lambda ST + (1 - \lambda)SA$ and $\mu DS + (1 - \mu)DN$, where

$$\lambda = \frac{r(ma + (1-m)b) - d}{(1-m)(rb - d)} \quad \text{and} \quad \mu = \frac{c}{b - kd}$$

both of which must be well-defined, and thus

$$a > d/k > b \quad \text{and} \quad b - kd > c.$$

must also be true. Furthermore, the condition

$$d > r(ma + (1 - m)b)$$

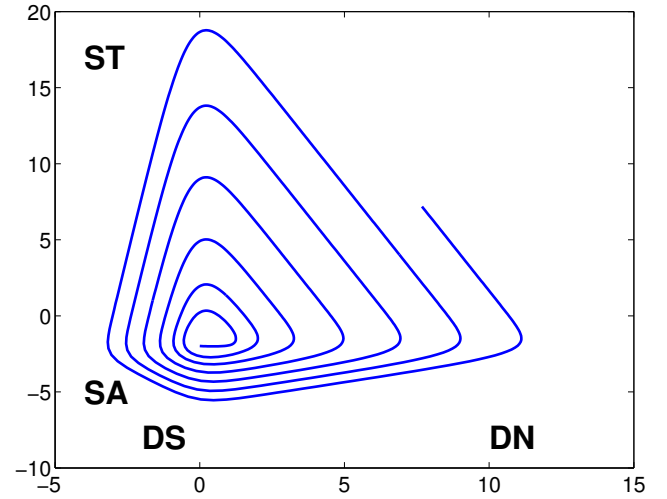


Figure 2: Example of a phase plot of strategies involved in hybrid equilibria. The evolutionary trajectory begins at the center of the spiral and moves outwards over time. X- and Y-coordinates denote the difference between the \log_{10} of the population fraction for the respective strategies.

is also required. An example of a hybrid equilibrium is shown in Figure 3. This evolutionary dynamic is reminiscent of the complex evolutionary dynamics which have been observed in continuous populations of Prisoner's Dilemma strategies (Lindgren, 1991). However, Lindgren's system eventually leads to an ESS, while hybrid equilibria spiral *ad infinitum* (Huttegger and Zollman, 2010).

Note that in the case of hybrid equilibria $b > 0$. This serves as an incentive for deceptive signaling, which is not a possibility in the case of Noble's game (Noble, 1999).

Population Dynamics

We evolve infinite populations with a two-population version of the discrete-time replicator equation (Sigmund and Hofbauer, 1998)

$$x_i(t+1) = \frac{\pi(Z(t), x_i)x_i(t)}{\sum_j \pi(Z(t), x_j)x_j(t)}$$

$$z_i(t+1) = \frac{\pi(X(t), z_i)z_i(t)}{\sum_j \pi(X(t), z_j)z_j(t)}$$

where $x_i(t)$ is the fraction of strategy i in the first population X at time t , $\pi(P, s)$ is the payoff of strategy s against population P , and $z_i(t)$ is the fraction of strategy i in the second population Z at time t . The fitness of a particular strategy is dependent upon the strategy distribution of the other population. This assumes complete mixing and that each strategy plays each other strategy.

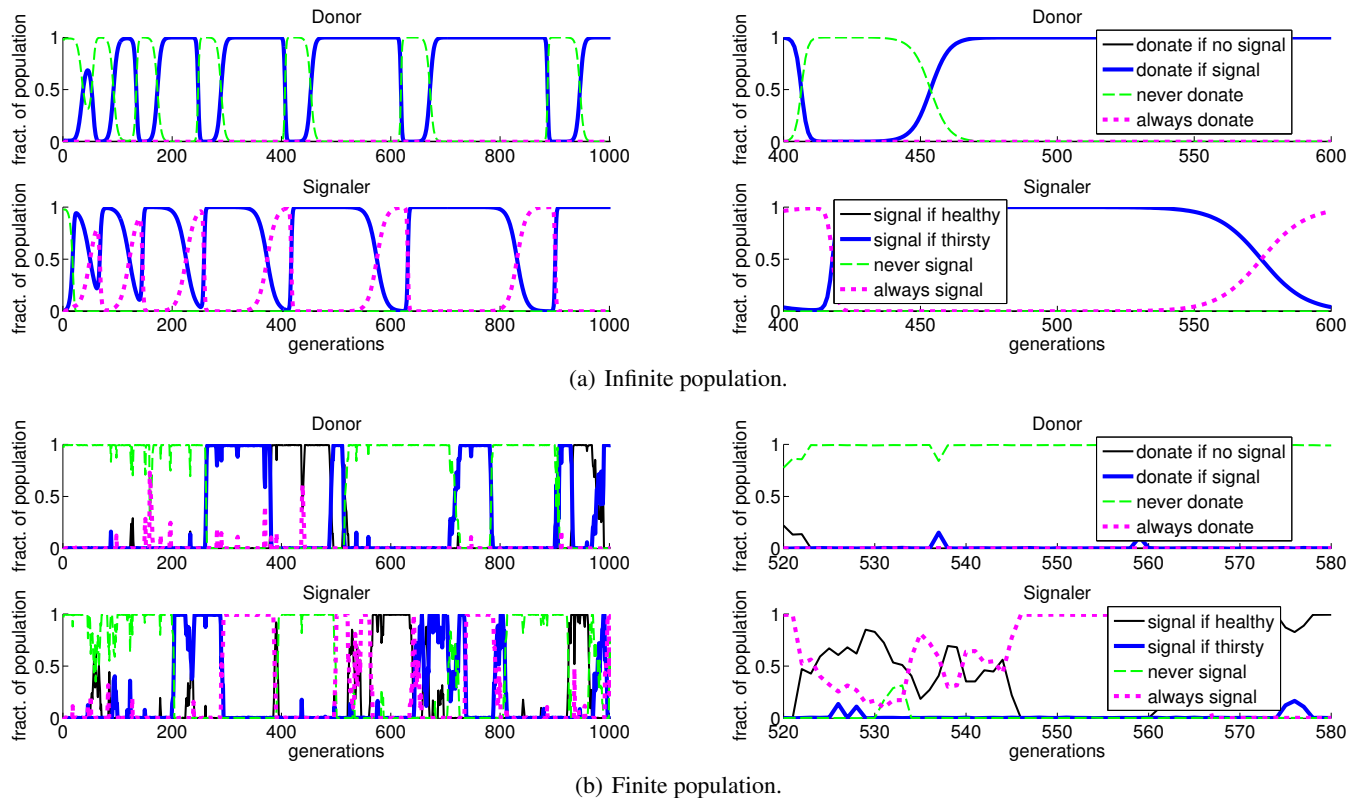


Figure 3: Example of a hybrid equilibrium. The graphs on the right hand side are zoomed in versions of those on the left. Note that the “signal if healthy” strategy invades the signaler population in the finite model. This strategy is essentially non-existent in the continuous model.

Evolutionary Algorithms

When evaluating finite populations we employ a simple genetic algorithm (Mitchell, 1996). In both populations individuals are represented as integers between 1 and 4 representing the strategies listed in Tables 1(c) and 1(b). Strategies are mutated with a probability of 0.01, and no crossover is used. Mutation is performed by replacing an individual with a randomly generated strategy. Each individual plays 50 games against randomly selected individuals from the opposing population, and the average payoff of these games is treated as the individual’s fitness.

A number of selection methods have been employed in evolutionary algorithms. In this study we focus on tournament selection due to its prevalence in the study of the emergence of signaling. In tournament selection, individuals are selected for reproduction by repeatedly choosing the best individuals from small randomly picked subsets. It has been shown that this “best-of-group” version of tournament selection has pathological behavior in terms of maintaining an ESS (Ficici, 2006). This finding helps motivate our hypothesis that this pathology might be present in studies of the emergence of signaling. (Nowak et al., 2004) extends the idea of ESS to finite populations as ESS_N where N is

the population size. We ensure that all individuals have an equal opportunity to compete by constructing tournaments with random permutations of the population.

Results

The results are presented in two sections. We first investigate the difficulty of reaching particular types of equilibria from non-communicative initial population distributions. We then use the parameters from the first investigation in a comparison of infinite and finite population sizes, the latter are investigated with multiple tournament sizes.

Emergence of Signaling

Game theoretic studies of the SPS game generally lead to statements about the existence of particular types of equilibria if certain conditions hold true for a given set of parameters. However, the existence of an equilibrium does not imply that the equilibrium is reachable from arbitrary population distributions. This has significant implications for the emergence of signaling. Under what conditions can an equilibrium be reached from a non-communicative origin?

We approach this question empirically. For each type of

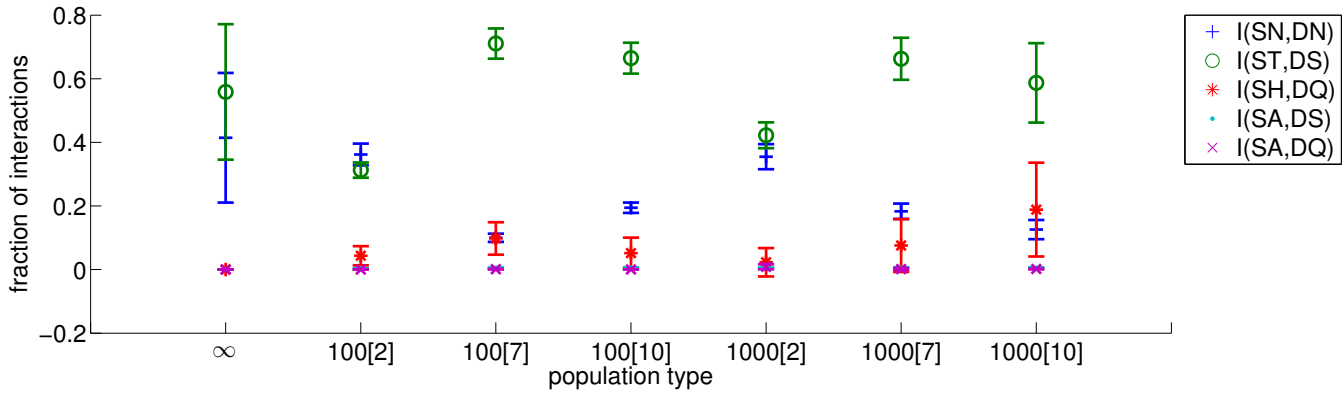


Figure 4: Results for honest signaling equilibria. $I(SX, DX)$, where SX and DX denote signaler and donor strategies, indicates the mean expected number of interactions over time with error bars showing standard deviation. “Infinite” identifies results from the continuous model. The rest of the labels in the form of $x[y]$, denote population and tournament size, respectively.

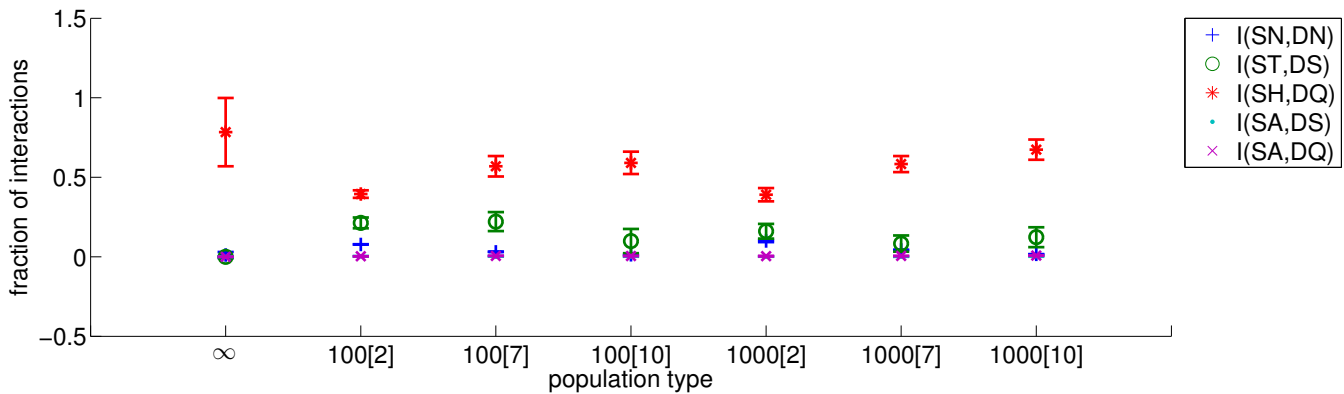


Figure 5: Results for inverse honest signaling equilibria.

equilibria we generate 1,000,000 random parameters¹ that satisfy the conditions presented in the sections describing equilibria, and test to see whether a continuous model initialized with non-communicative population distributions actually reaches the target equilibrium. The success rate for a given equilibria type quantifies the size of the basin of attraction in parameter space.

Populations are initialized with primarily non-signalers and non-donors (97% of the population) and small fractions of the remaining strategies (1%). We evolve the populations with the discrete-time replicator for 1,000 generations and test to see if the evolutionary trajectory matches that of the corresponding equilibria. For signaling and noncommunicative equilibria, we assume that the system has reached the target if the dominant strategy for signalers and donors matches that of the given equilibrium. For hybrid and pooling equilibria, we compute the mean of the distribution of strategies over time. We look for a match using these means

¹While 1,000,000 may seem like a large number of parameters to test, evaluations of the continuous model are very fast.

for dominant signaler and donor strategies, assuming that strategies with continuously small distributions are eliminated. All parameters that produce the appropriate behavior within 1,000 generations are recorded. In Table 2 we present the success rate for reaching particular equilibria from non-communicative initial conditions.

We can see that honest signaling, followed by hybrid, are the hardest type of equilibria to reach given noncommunicative population distributions. This is followed by inverse honest signaling and pooling II (where donor strategies are a mix of DA and DQ against SN) equilibria. We observe that of the 1,000,000 parameter sets generated for each, less than 10% were able to reach any of these target equilibria. It is not particularly intuitive that inverse honest signaling equilibria are easier to reach than the honest signaling equilibria. However, we note that in order to reach an inverse honest signaling equilibrium the system must pass through a configuration like that of a pooling equilibrium. The pooling equilibrium that it passes through is the SN and DA/DQ profile. Additionally, it can be seen that it is easier to reach an hybrid equilibrium than an honest signaling equilibrium

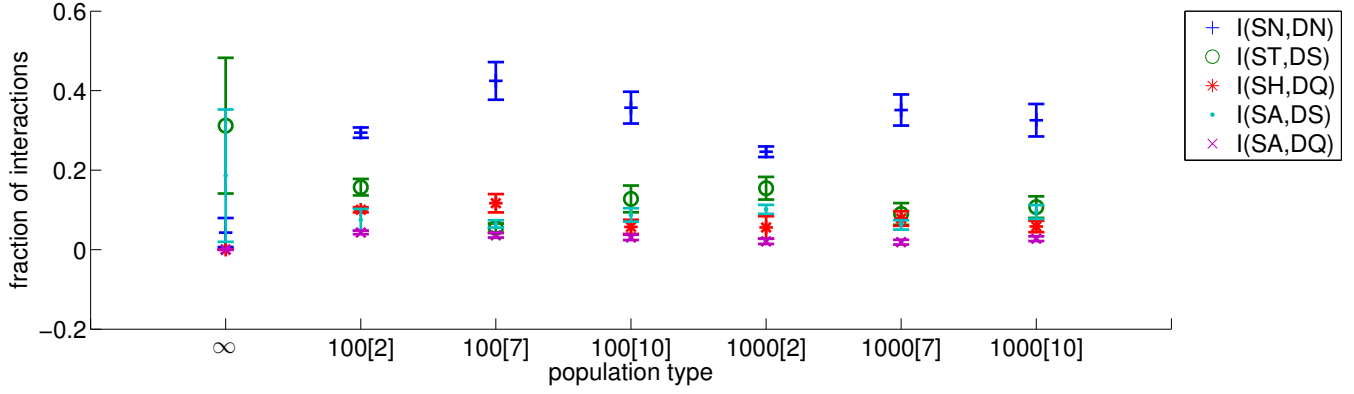


Figure 6: Results for hybrid equilibria.

Equilibrium type	Success rate
Honest signaling	0.0027
Inverse honest signaling	0.0628
Pooling I	0.9996
Pooling II	0.0969
Hybrid	0.0129

Table 2: Success rate for reaching the appropriate equilibrium from non-communicative initial conditions. Rates are computed based upon 1,000,000 randomly generated parameters that satisfy the conditions of the respective equilibria.

from non-communicative initial conditions.

Infinite and Finite Populations

We are interested in the emergence of signaling, as such all simulations are initialized with populations of primarily non-signalers and non-donors. The populations are evolved for 1,000 iterations for both infinite and finite populations. For each equilibria, 200 parameter sets are randomly chosen from those that reached target in the previous search. Then each evolutionary configuration is evaluated on a given parameter set. Finite populations are repeated 10 times and averaged.

We measure the distance from the true equilibrium with the expected number of interactions given the current population distributions. This is denoted as

$$I(SX, DX) = \frac{|SX| * |DX|}{\sum_{Si, Dj} |Si| * |Dj|}$$

where SX is the signaler strategy of interest, DX is the donor strategy of interest, $Si \in \{ST, SH, SA, SN\}$, and $Dj \in \{DS, DQ, DA, DN\}$.

We take the mean for each expected interaction over time for both the continuous and finite models. For finite populations we look at population sizes of 100 and 1,000, and tour-

namment sizes of 2, 7, and 10. These population sizes span the order of magnitudes that are generally used in studies of the emergence of signaling. Likewise, these tournament sizes span the range commonly used in such studies.

Figures 4 and 5 suggest that the finite model is a good approximation of the continuous model for Nash equilibria. The expected interactions for finite populations of both sizes roughly estimate those calculated in the continuous model (labeled infinite on the x-axis) for tournament sizes greater than 2. Figure 4 suggests that the finite populations approach the behavior of the infinite population as tournament size increases. Tournaments of size 2 perform particularly poorly relative to bigger tournament sizes in the case of signaling equilibria. This is counter to Reggia et al.'s finding where they see that smaller tournament sizes actually lead to higher proportions of signalers in the population (Reggia et al., 2001). This leads us to suggest that in their case the complex environment and agent architecture may have a bias towards signaling behavior.

In Figure 6 we see that the finite model fails to capture the complex dynamics of hybrid equilibria. This is because hybrid equilibria are actually collections of polymorphic mixed Nash equilibria. It has previously shown that tournament selection cannot converge to polymorphic Nash equilibria in both one- (Ficici et al., 2005) and two-population coevolution (Ficici, 2006). This leads us to question the significance of the dynamics observed in previous studies of the emergence of signaling. If it is not possible for a simple evolutionary model with tournament selection to maintain a polymorphic Nash equilibrium, then what are the complex dynamics that have previously been observed (Bullock, 1997)? We suggest that these types of dynamics may be a direct result of the spatial selection mechanism based upon previous findings that spatial games can produce behaviors ranging from chaotic dynamics to asymptotically predictable population dynamics (Nowak and May, 1992; Roca et al., 2009).

Conclusion

We have presented a coevolutionary study of the effects of evolutionary mechanics on the emergence of signaling. In doing so we quantify Bullock's previous finding that the existence of a signaling equilibrium does not imply that it can be reached from an initially non-communicative state (Bullock, 1997). It is also shown that it is significantly easier for signaling to evolve from non-communication to an inverse signaling equilibrium than to the signaling equilibrium traditionally studied in the SPS game. Recall that the difference between these two signaling equilibria is when the signal is sent, while the donor adopts the response corresponding to honest signaling. This observation aligns with the signal of the peacock's tail to the peahen, which is a demonstration of virility not aridity.

Finally, we have shown that finite population models with tournament selection can fail to capture the dynamics of hybrid equilibria, one of the most attractive dynamics of the SPS game. These equilibria (which are actually families of polymorphic Nash equilibria) follow a spiraling trajectory that switch between honest and deceptive signaling. The inability of tournament selection to maintain polymorphic Nash equilibria is already known (Ficici et al., 2005). The enhanced reachability of hybrid equilibria relative to traditional signaling ESS's suggests that the generalizability of evolutionary models which fail to capture this phenomenon are limited.

Acknowledgments

We thank the DEMO lab, past and present; in particular, Sevan Ficici. Computing support was provided by the Brandeis HPC.

References

- Bergstrom, C. (2012). The Theory of Honest Signaling. http://octavia.zoology.washington.edu/handicap/honest_biology_01.html.
- Bergstrom, C. and Lachmann, M. (1997). Signalling among relatives. I. Is costly signalling too costly? *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 352(1353):609–617.
- Bullock, S. (1997). An exploration of signalling behaviour by both analytic and simulation means for both discrete and continuous models. In *Proceedings of the Fourth European Conference on Artificial Life*, pages 454–463.
- Ficici, S. (2006). A game-theoretic investigation of selection methods in two-population coevolution. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 321–328.
- Ficici, S., Melnik, O., and Pollack, J. (2005). A game-theoretic and dynamical-systems analysis of selection methods in coevolution. *Evolutionary Computation, IEEE Transactions on*, 9(6):580–602.
- Ficici, S. and Pollack, J. (2007). Evolutionary dynamics of finite populations in games with polymorphic fitness equilibria. *Journal of Theoretical Biology*, 247(3):426–441.
- Fogel, D. and Fogel, G. (1995). Evolutionary stable strategies are not always stable under evolutionary dynamics. *Evolutionary Programming IV*, pages 565–577.
- Grafen, A. (1990). Biological signals as handicaps*. *Journal of Theoretical Biology*, 144(4):517–546.
- Hamilton, W. (1964). The genetical evolution of social behaviour. II. *Journal of Theoretical Biology*, 7(1):17–52.
- Hurd, P. (1995). Communication in discrete action-response games. *Journal of Theoretical Biology*, 174(2):217–222.
- Huttenberger, S. and Zollman, K. (2010). Dynamic stability and basins of attraction in the Sir Philip Sidney game. *Proceedings of the Royal Society B: Biological Sciences*, 277(1689):1915–1922.
- Johnstone, R. and Grafen, A. (1992). The continuous Sir Philip Sidney game: a simple model of biological signalling. *Journal of Theoretical Biology*, 156(2):215–234.
- Lindgren, K. (1991). Evolutionary phenomena in simple dynamics. *Artificial Life II*, 10:295–312.
- Maynard Smith, J. (1982). *Evolution and the Theory of Games*.
- Maynard Smith, J. (1991). Honest signalling: The Philip Sidney game. *Animal Behaviour*, 42:1034–1035.
- Maynard Smith, J. and Price, G. R. (1973). The Logic of Animal Conflict. *Nature*, 246(5427):15–18.
- Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. MIT Press.
- Noble, J. (1999). Cooperation, conflict and the evolution of communication. *Adaptive Behavior*, 7(3-4):349–369.
- Nowak, M. and May, R. (1992). Evolutionary games and spatial chaos. *Nature*, 359(6398):826–829.
- Nowak, M., Sasaki, A., Taylor, C., and Fudenberg, D. (2004). Emergence of cooperation and evolutionary stability in finite populations. *Nature*, 428(6983):646–650.
- Ozisk, A. P. and Harrington, K. I. (2012). The Effects of Tags on the Evolution of Honest Signaling. In *ECOMASS Workshop, Workshop Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2012)*, in press.
- Reggia, J., Schulz, R., Wilkinson, G., and Uriagereka, J. (2001). Conditions enabling the evolution of inter-agent signaling in an artificial world. *Artificial Life*, 7(1):3–32.
- Roca, C., Cuesta, J., and Sánchez, A. (2009). Evolutionary game theory: Temporal and spatial effects beyond replicator dynamics. *Physics of Life Reviews*, 6(4):208–249.
- Saunders, G. and Pollack, J. (1996). The Evolution of Communication Schemes Over Continuous Channels. In *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pages 580–589.
- Sigmund, K. and Hofbauer, J. (1998). Evolutionary games and population dynamics. *Cambridge Univ. Press*.
- Wagner, K., Reggia, J., Uriagereka, J., and Wilkinson, G. (2003). Progress in the simulation of emergent communication and language. *Adaptive Behavior*, 11(1):37–69.
- Zahavi, A. (1975). Mate selection—a selection for a handicap. *Journal of Theoretical Biology*, 53:205–214.

App Epidemics: Modelling the Effects of Publicity in a Mobile App Ecosystem

Soo Ling Lim¹ and Peter J. Bentley²

¹Department of Statistical Science

²Department of Computer Science

University College London, United Kingdom

s.lim@cs.ucl.ac.uk

Abstract

In mobile app ecosystems, an app can behave like a virus. Once downloaded, it may cause its user to recommend that app to friends who then may download the app and “infect” other friends. Epidemics occur when a small number of downloads causes a snowballing effect that results in a massive number of downloads (and consequently, a rich developer). This paper presents AppEco, the first Artificial Life model of mobile application ecosystems. AppEco models the app store, app developers, apps, users, and their behaviour. We use AppEco to simulate Apple’s iOS app ecosystem and investigate common publicity strategies adopted by developers and their effects on app downloads. Specifically, we investigate three causal factors for a widespread “app infection” from epidemiology: the users’ exposure to the app, the users’ susceptibility to the app, and the infectiousness of the app.

Introduction

In our technological world we frequently mirror the natural world, often without realising it. Today we have “mobile app ecosystems”, in which app developers build software apps and users consume the apps in an environment provided by an app store. Within the app ecosystem, developers may evolve their strategies, adapting to the requirements of users and producing new apps that fit into ever-changing niches. Apps may infect users like a virus – once downloaded, an app may cause its user to recommend that app to friends who then may download the app, and so on. Consequently, while epidemiologists develop new ways to prevent the spread of biological viruses, app developers are trying to find the most effective ways to spread their apps virally.

In this study, we use knowledge from the field of epidemiology to investigate the effectiveness of common app publicity strategies. One approach to such a study might be to experiment with a real app store: flood the store with thousands of new apps, publicise them using different strategies, and analyse the results. However, some strategies, such as television broadcasting of the app to millions of users, are costly to implement in the real world. Other strategies, such as paying users to download an app in order to manipulate its ranking on the app store, are frowned upon¹. Another approach might be to attempt to use machine learning

to predict success or failure based on past data. However, data on publicity for specific apps and resulting downloads is not available, and because the app store comprises a non-static (constantly-growing) complex system, such predictions could never be made with any confidence. For these reasons, we use an Artificial Life (Alife) agent-based model as an experimental tool for this work. Alife methods have proven their worth with many previous simulations of ecosystems.

In this paper, we present AppEco, an Alife model of mobile app ecosystems. AppEco models developers (agents that build apps) and users (agents that download apps). It simulates the app store environment, which hosts and organises apps, and enables users to browse and download apps. Significantly, AppEco also models apps (artefacts produced by the developers and downloaded by users) and their features. AppEco allows us to conduct experiments, test hypothesis about various processes in the ecosystem, and ask “what if” questions, all of which are difficult if not impossible to conduct in a real-world setting. We use AppEco to simulate Apple’s iOS app ecosystem and study common publicity strategies adopted by developers and their effects on app downloads. Specifically, we investigate three causal factors for a widespread “app infection” from epidemiology: the users’ exposure to the app, the users’ susceptibility to the app, and the infectiousness of the app.

The rest of the paper is organised as follows. The following section describes existing work. The section after that describes AppEco. We then describe the application of AppEco to simulate the iOS app ecosystem, the experiments and results. The final section provides our conclusions.

Background

Epidemiology is the study of the distribution and determinants of diseases and other health-related events in specified populations, and the application of this study to the control of health problems (Dicker et al., 2006). Much epidemiologic research is devoted to searching for causal factors that influence one’s risk of disease so that appropriate public health action might be taken (Rothman et al., 2008). A simple model of disease causation for infectious disease is the epidemiologic triangle, which consists of an external agent, a susceptible host, and an environment that brings the host and agent together. Disease results from the interaction between the agent and the susceptible host in an environment that

¹ <http://paidcontent.org/article/419-apple-promises-a-crack-down-on-those-who-manipulate-app-store-rankings/>

supports transmission of the agent from a source to that host (Dicker et al., 2006). An epidemic is the occurrence of more cases of disease than expected in a given area or a specific population during a particular period. Epidemics occur when an agent and susceptible hosts are present in adequate numbers, and the agent can be effectively conveyed from a source to the susceptible hosts. The chances of an epidemic increases when there is an increase to host susceptibility or an increase to host exposure (Dicker et al., 2006).

In the fields of Alife, Evolutionary Computing, and Agent-Based Simulation, researchers have modelled various aspects of ecosystems such as evolutionary dynamics within interacting populations. Classic works in this area include studies by Axelrod and Hamilton (1981) on the evolution of cooperation and Maynard Smith and Price (1973) on conflicts between animals of the same species. More recently, Holland (1992) created Echo, a generic ecosystem model in which evolving agents are situated in a resource-limited environment. Olague et al. (2006) developed the infection algorithm, a bio-inspired approach based on an artificial epidemic process, to address the stereo image matching problem. Dorin (2005) used knowledge from epidemiology to model the co-evolution of transmissible disease and a population of non-randomly mixed susceptible agents. Agar and Wilson (2002) used simulation to study illicit drug epidemics. App stores have large populations of apps, developers, and users, and can benefit from similar studies.

While the study of mobile app ecosystems is a current and significant topic for researchers, to date there has been little work focussing on app publicity (Jansen et al., 2009; Lin & Ye, 2009). However there is related work that contextualises and informs our study. For example, Garg and Telang (2011) developed algorithms to predict the current sales of an app based on its ranking on Apple's iOS App Store Top Apps Chart. Such work may enable investors to estimate likely profits should an app reach a specific rank, however there is no certainty that a new app will appear on the chart. Bohmer et al. (2011) developed a mobile app to collect mobile app usage information from over 4,100 users of Android devices. They found that although users spend almost an hour a day using their phones, an average session with an app lasts less than a minute. They also found that news apps are most popular in the morning and games are at night, but communication apps dominate through most of the day. These studies are informative, but limited to studying what is already out there, and "what-if" questions cannot be answered.

AppEco

In an app ecosystem, coevolving systems of apps, developers, and users form complex relationships, filling niches, competing and cooperating, similar to species in a biological ecosystem (Lin & Ye, 2009). The health of the app ecosystem is largely determined by the communities of developers that create innovative solutions that users want to buy (Cusumano, 2010; Jansen et al., 2009). In an app ecosystem, application software (such as games, medical applications, and productivity tools) that is built for a mobile platform is sold via an app store running on the platform. The app store concept has democratised the software industry – almost anyone can build and sell apps. Once built, an app quickly

becomes available to a worldwide market. Mobile device users can download the apps, use them immediately and provide feedback to the developers.

AppEco is an Artificial Life simulation of mobile app ecosystems. The model consists of agents that are abstractions of app users and developers, as well as artefacts that are abstractions of apps. Developer agents build and upload apps to the app store; user agents browse the store and download the apps, see Figure 1. Each download corresponds to a new sale. A distinguishing feature of the AppEco model compared to more traditional agent-based models is the explicit modelling of artefacts as well as the agents that produce and use the artefacts. Different from agents, artefacts are not autonomous, they represent passive entities of the system that are intentionally created and used by agents. App artefacts are important in a model of an app ecosystem because the agents interact with one another via the apps. An earlier version of AppEco is described in Lim and Bentley (2012).

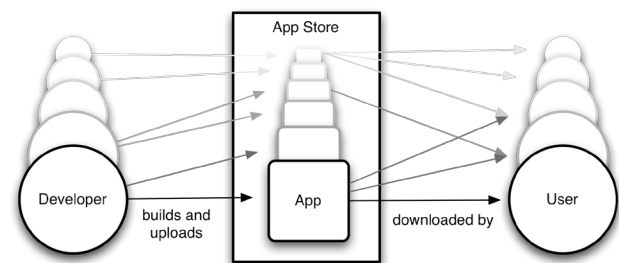


Figure 1. The interaction between developers, apps, and users in AppEco (Lim & Bentley, 2012).

AppEco Components

AppEco consists of app developers, apps, users, and the app store. Each component is described as follows.

Developers. In AppEco, a developer agent represents a solo developer or a team of developers working together to produce an app. Each developer agent has a development duration (*devDuration*, a random value between [dev_{min} , dev_{max}]), which specifies the number of days it needs to build an app. Each developer also records the number of days it has already spent building the app (*daysTaken*). Each developer is initially active (it continuously builds and upload apps to the app store) but may become inactive (it stops building apps) with probability $P_{inactive}$. This models part-time developers, hobbyists, and the tendency of developers to stop building apps². Every developer records the number of apps it has developed and the number of downloads it has received.

In this work every developer uses an evolutionary strategy of making a variation of its own best app (app with highest number of downloads) each time. This models the ability of developers to learn from downloads and improve on their best app. This strategy is commonly used by developers who learn from their experience. An example is Rovio, who developed many game apps before hitting the jackpot with Angry Birds. They then built on their success, releasing new apps such as

² <http://t-machine.org/index.php/2009/06/11/may-2009-survey-of-iphone-developers/>

Angry Birds Seasons, and Angry Birds Rio³. (Our previous work investigated developers with different development strategies (Lim & Bentley, 2012). For example, innovative developers build a different app each time; copycats copy other developers' apps. Here we use the evolutionary strategy alone to simplify analysis; the effects of publicity are not significantly altered by the use of different developer strategies.)

Apps. Each app artefact is built and uploaded by a developer agent. The features of the app are abstracted as a 10x10 feature grid (**F**) for each app. If a cell in **F** is filled, then the app offers that particular feature. A grid is used so that feature similarity can be represented in the future, e.g., features that are similar can be represented as cells that are near to one another on the grid. The cells in **F** are filled probabilistically if this is the developer's first app. Otherwise, the developer fills **F** with copies of the features from his own best app (as determined by the highest daily average downloads) with random mutation. The choice of which app to copy occurs when the developer is starting to build the app. If no apps by this developer have downloads, the developer fills **F** with a copy of his most recent app. There is a 0.5 probability that mutation occurs during a copy. Mutation is implemented by randomly selecting a filled cell in **F** and randomly "moving" it to an empty cell in **F**.

For ranking purposes, each app keeps a record of the total number of downloads it has received to date and the number of downloads it has received on each of the previous seven days. Each app has a probability of $P_{\text{Infectious}}$ to be infectious. If the app is infectious, users who download the app recommend it to their friends. Apps can be infectious because they have exciting features (e.g., Angry Birds). Apps can also have infectious features. For example, WhatsApp Messenger⁴ (No. 1 in 99 countries) is a mobile messaging app that allows users to exchange messages without having to pay for SMS. The user needs his friends to download the app to receive his messages. His friends will, in turn, ask their friends to download the app. For simplicity, the AppEco model currently assumes that all apps are sold at the same price; the model of variations in app pricing and categories of apps is left for future work. Each app also records the time when it was uploaded.

Users. Inspired by the recommender systems literature (Adomavicius & Tuzhilin, 2005), each user agent has preferences (or taste information) that determine the app features that it prefers. Developers are unaware of the users' preferences. The preferences of a user agent are abstracted as a 10x10 preference grid (**P**). The top right quadrant in **P** is always empty, to model features that are undesirable to all users. For example, no users want an app to have the features of a difficult-to-use or malicious program. The top left and bottom right quadrant in **P** are filled probabilistically, such that each cell in the grid has a probability P_{Pref} of being filled, to model features that are desirable to some users. The bottom left quadrant in **P** is filled probabilistically, such that each cell

in the grid has a probability $2 \times P_{\text{Pref}}$ of being filled, to model popular features desirable to many users. An example preference grid is illustrated in Figure 2 (right).

If a cell in **P** is filled, then the user agent desires the feature represented by that cell. If the feature grid **F** of an app has a cell in the same location filled, then it means the app offers a feature desired by the user agent (i.e. the user is susceptible to infection by that app). For example, in Figure 2, all four of the features offered by App 1 match the user agent's preferences, but only two of the features offered by App 2 match the user agent's preferences. Using the AppEco model, an app such as Angry Birds (to which many users are susceptible) can be abstracted as an app with **F** that matches **P** of many users, while an app to which few users are susceptible has **F** that matches few or no users' **P**. For simplicity, preference matching is binary: filled cells either match or do not match.

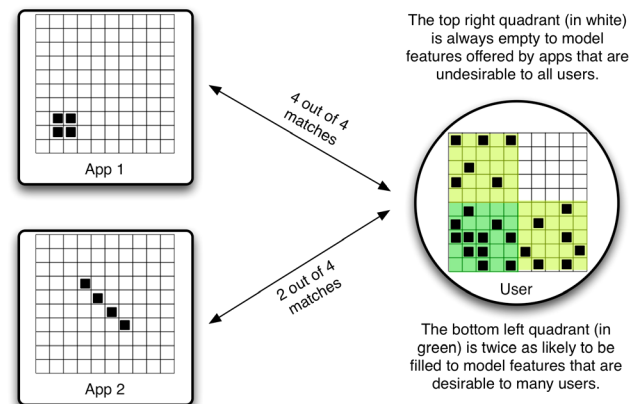


Figure 2. Matching app features with user preferences.

Each user agent keeps a record of the apps it has downloaded, the number of days between each browse of the app store (*daysBtwBrowse*, a random value between $[\text{bro}_{\text{min}}, \text{bro}_{\text{max}}]$), and the number of days that have elapsed since it last browsed the app store (*daysElapsed*). *daysElapsed* is recorded so that the user agent knows when to browse the app store next. When users are initialised at the start of the simulation, *daysElapsed* is set to be a random number between $[0, \text{daysBtwBrowse}]$ so that users don't all browse at the same time when they start. Users also record the number of friends they can influence and thus potentially "infect" (*numFriends*). The value of *numFriends* is a random number with a power law distribution in the range $[0, 150]$. (Many people will be able to influence very few friends, but a few people can influence many friends.) The upper limit of this range is derived from the Dunbar number of 150 (Dunbar, 1992). Dunbar (1992) showed that the human brain is only capable of managing relationships with about 150 people (staying in contact at least once per year and knowing how friends relate to others). Dunbar suggests that this number remains the same despite new social networking technologies such as Facebook and Twitter⁵.

³<http://www.wired.co.uk/magazine/archive/2011/04/features/how-rovio-made-angry-birds-a-winner>

⁴ <http://www.whatsapp.com/>

⁵http://www.nytimes.com/2010/12/26/opinion/26dunbar.html?_r=2&ref=facebookinc

App Store. The app store is the environment used by the agents to store and access apps. Its primary function is to provide a shop front for users and enable them to locate and download apps that match their preferences. To achieve this, it provides three browsing methods: the Top Apps Chart, the New Apps Chart, and Keyword Search. These browsing methods provide changing subsets of apps to users; they are the “watering holes” of the ecosystem at which all users drink. As such, they provide a vital mode of transmission of apps to users. These three methods are modelled because they are common to many app stores, such as iOS, Android, and BlackBerry. The Top Apps Chart ranks apps based on the number of downloads the apps have received. The New Apps Chart displays apps that have recently been uploaded by developer agents; only a small subset of new apps is chosen for the chart. Keyword Search returns a list of apps that match the keyword entered by the user agent. In AppEco, Keyword Search is abstracted as a random search for a random number of apps. It is implemented in this way because keywords may not correspond to features, so a matching keyword does not mean the app has desirable features for the user.

AppEco Algorithm

The AppEco algorithm models the daily interactions between the AppEco components described in the previous section. AppEco is implemented in C++. Each timestep in the algorithm represents a day in the real ecosystem.

Inspired by the ecology literature (Kingsland, 1995), the population growth of user and developer agents is modelled using a sigmoid growth function commonly used to model the population growth in natural systems. The equation models the growth rate of user and developer agents in an app ecosystem declining as their population density increases, with the size of the ecosystem limited by the market share of the mobile platform. The population size at timestep t , pop_t , is defined by Equation 1.

$$pop_t = \text{MinPop} + \frac{(\text{MaxPop} - \text{MinPop})}{1 + e^{S(t-D)}} \quad (1)$$

where MinPop is the minimum population, MaxPop is the maximum population, S determines the slope of the growth curve (S is negative for a growth curve), and D shifts the curve from left to right. Different growth formulas (Kingsland, 1995) can be used to model different ecosystems.

The AppEco algorithm is depicted in Figure 3 and detailed as follows.

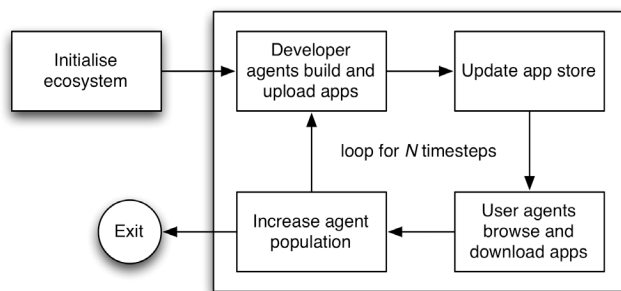


Figure 3. The AppEco algorithm.

Initialise ecosystem. This step launches AppEco with the population of developer and user agents as defined in Equation 1, with timestep $t = 0$. It is common for app stores to have apps before it is opened. For example, the iOS App Store had 500 apps the day it was launched⁶. As such, this step also creates an initial number of app artefacts (N_{InitApp}). The developers of these initial apps are randomly selected from the pool of initial developers. The attributes of initial developers, apps, and users are set as described in the previous section.

Developer agents build and upload apps. For each active developer, *daysTaken* is incremented by 1. If *daysTaken* exceeds this developer’s *devDuration*, the app is completed. The developer then uploads the app to the store, resets *daysTaken* to 0. The *feature* attribute of the app is set such that each cell in the 10x10 feature grid has a probability P_{Feat} of being filled.

Update app store. The New Apps Chart is updated. When timestep $t = 0$, the New Apps Chart consists of a random selection of initial apps. In each following timestep, each new app has a probability $P_{\text{OnNewChart}}$ of appearing on the New Apps Chart. Apps are randomly selected here because the selection criteria are not the focus of this work and real app stores do not reveal how they select apps for the New Apps Chart. The maximum number of apps in the chart is defined by $N_{\text{MaxNewChart}}$. As newly selected apps are added to the chart, older apps appear lower in the chart and are no longer listed when their position exceeds the chart size. The Top Apps Chart is also updated. When timestep $t = 0$, the Top Apps Chart is empty because no apps have been downloaded yet. In each following timestep, apps are ranked in the order of decreasing score, calculated as $8*D_1 + 5*D_2 + 5*D_3 + 3*D_4$ where D_n is the number of downloads received by the app on the n th day before the current day⁷. The maximum number of apps in the Top Apps Chart is defined by $N_{\text{MaxTopChart}}$.

User agents browse and download apps. For each user, *daysElapsed* is incremented by 1. If *daysElapsed* exceeds *daysBtwBrowse*, then the user browses the app store and resets *daysElapsed* to 0. The user browses the New Apps Chart and the Top Apps Chart, and conducts Keyword Search (which returns a random number of apps between [key_{min} , key_{max}]). The user browses each app that it has not previously downloaded: the feature grid of the app is compared with the preference grid of the user. If all the features offered by the app match the user’s preferences, then the user downloads the app. For example, in Figure 2, the user downloads App 1 but not App 2. If the user has downloaded an infectious app in the current timestep, the user will recommend the app to his friends who will then browse the app in the next timestep and download the app if it matches their preferences.

Increase agent population. This step increases the number of user and developer agents in the ecosystem for the next timestep, using Equation 1.

⁶ <http://www.apple.com/pr/library/2008/07/10iPhone-3G-on-Sale-Tomorrow.html>

⁷ <http://www.slideshare.net/misteroo/how-to-market-your-app>

Experiments

It is the dream of all developers to be able to spread their app throughout an ecosystem and have their apps “infect” as many users as possible. We first calibrate AppEco to match, as much as is feasible, the behaviour of a real app store. We selected Apple’s iOS App Store for our experiments, as it is one of the oldest and most established app stores. We then perform experiments to investigate three causal factors for epidemics that are relevant to mobile app ecosystems (Dicker et al., 2006).

Calibrating AppEco for iOS

We collected the following iOS data over a period of three years, from the start of the iOS ecosystem in July 2008 (Q4 2008) until the end of June 2011 (Q3 2011):

- **Number of iOS developers.** The number of iOS developers is based on the number of worldwide iOS developers month over month compiled by Gigaom⁸.
- **Number of iOS apps and downloads.** The number of apps and downloads is based on statistics provided in Apple press releases and Apple Events⁹. For example, in the Apple Special Event on 9th Sept 2009, Apple CEO Steve Jobs announced the App Store to reach 75,000 apps and 1.8 billion downloads, and Apple’s press release on 28th Sept 2009 announced that the App Store had more than 85,000 apps and 2 billion downloads¹⁰.
- **Number of iOS users.** The number of iOS users is based on the number of iOS devices (iPod Touch, iPhone, and iPad) sold by Apple over time. The sales figures are available from Apple’s quarterly financial data¹⁰, and for simplicity the calculation assumes that each user has one iOS device.

Using this and other publically available data we calibrated AppEco to simulate the iOS app ecosystem. Table 1 summarises the calibrated values for the system constants. In order to match (curve-fit) the iOS user and developer growth rates, values such as D and S for users and developers were determined through tuning experiments.

[Pop _{min} User, Pop _{max} User]	[1500, 40000]	[dev _{min} , dev _{max}]	[1, 180]
D _{User}	-4.0	P _{Pref}	0.4
S _{User}	-0.0038	P _{Feat}	0.04
[Pop _{min} Dev, Pop _{max} Dev]	[1000, 120000]	P _{OnNewChart}	0.001
D _{Dev}	-4.0	N _{MaxNewChart}	40
S _{Dev}	-0.005	N _{MaxTopChart}	50
N _{InitApp}	500	P _{Inactive}	0.0027
[bro _{min} , bro _{max}]	[1, 360]	[key _{min} , key _{max}]	[0, 50]

Table 1. Constant Values Resulting from iOS Calibration

It is computationally infeasible in terms of memory to simulate hundreds of millions of users. To ensure that the system is computationally feasible, one app represents one real app, and one developer agent represents one real

developer, but one user agent represents 10,000 real users. As such, the value of *numFriends* for one user agent is the average *numFriends* for 10,000 real users. This coarse-grain simulation is necessary to enable the modelling of the entire app ecosystem using the available computing resources. Mobile app ecosystems are international ecosystems. Publicity occurs in different countries and infections are not bounded by the users’ physical location. For this reason, modelling just one country or a subset of app users would not provide an accurate simulation of the true app store ecosystem.

After calibration the behaviour of AppEco closely resembles the behaviour of the iOS ecosystem, including emergent rates such as the number of apps and downloads. A run of the simulation takes approximately 22 seconds CPU time on a MacBook Air with a 1.8GHz Intel Core i7 Processor and 4GB of 1333 MHz DDR3 memory. After three years (1080 timesteps assuming 30 days a month), the model typically contains more than 100,000 developer agents, 500,000 apps, 20,000 user agents (corresponding to 200m real users), and 1.5 million downloads (corresponding to 15bn real downloads).

Experimental Setup

Our objective in the experiments is to understand the effects of app publicity in the ecosystem: what makes an app epidemic? We investigate three causal factors for epidemics (Dicker et al., 2006):

- **Changes in host exposure**, through app publicity and app appearance on app store charts
- **Changes in host susceptibility**, by varying the degree to which app features meet the preferences of users, and
- **Changes in app infectiousness**, by varying whether users influence their friends to download the app. Although epidemics usually result from infectious agents, non-infectious diseases can also exist in epidemic proportions.

In each experiment, an app is inserted into the App Store two years after the App Store is opened (timestep = 720), when the ecosystem is reasonably mature. The app is then studied for a year. We model *changes in host exposure* by using the following publicity strategies for the app:

- **No exposure.** The app is not publicised.
- **Mass exposure.** A total of 100 user agents (corresponding to 1,000,000 real users) look at the app. This models mass broadcast such as TV and radio¹¹. If the users like the app’s features, they will download the app and if the app is infectious, they will recommend the app to their friends, as described in the previous section.
- **Targeted exposure.** One targeted user agent (10,000 real users) will look at the app. This models advertisements through specialist magazines or conferences to influential users⁷. Targeted users are users whose preferences matches the app’s features (chosen by selection from a random sample of 1000 users agents). The targeted user will recommend the app to his friends regardless of whether the app is infectious.

⁸ <http://gigaom.com/apple/infographic-apple-app-stores-march-to-500000-apps/>

⁹ <http://www.apple.com/apple-events/>

¹⁰ <http://www.apple.com/pr/library/>

¹¹ For example, <http://www.bbc.co.uk/news/11145583>

- **Recurring exposure.** One user agent (10,000 real users) looks at the app, first at timestep T_{AppPub} , and then after that a different user agent will look at the app every 30 timesteps, for 6 times. This models periodical advertisements through magazines or websites⁷.
- **Enhancing mode of transmission through Top Apps Chart (TAC).** Some developers choose to pay people to download their apps in order to inflate artificially the total number of downloads the app receives¹. Their goal is to improve the app's ranking on the Top Apps Chart, which will then make the app more visible to users and increase subsequent downloads. We model this strategy by adding one unit of downloads (equivalent to 10,000 downloads in the real world) to an app.
- **Enhancing mode of transmission through New Apps Chart (NAC).** Similar to the Top Apps Chart, appearing on the New Apps Chart also increases user visibility. This chart is known as the New and Noteworthy Chart on the iOS App Store. Although the selection criteria for the chart is unclear, developers who have succeeded suggested making innovative and desired apps, creating new interfaces for existing app features¹², and most commonly, getting the app known by the right people¹³. We model this strategy by placing the app at the top of the New Apps Chart on a given timestep.

We model *changes in host susceptibility* by designing three apps for the experiments: an “excellent” app with features only in the bottom left of **F**, ensuring that a large percentage of users will be susceptible to infection, a “good” app with two features in the bottom left of **F** and two features in the top left and bottom right quadrants of **F**, and an “average” app with features in the top left and bottom right quadrants of **F**, as illustrated in Figure 4. Bad apps with features in the top right quadrant of **F** will never be downloaded regardless of publicity. For this reason experiments with bad apps are entirely predictable and are excluded from the study.

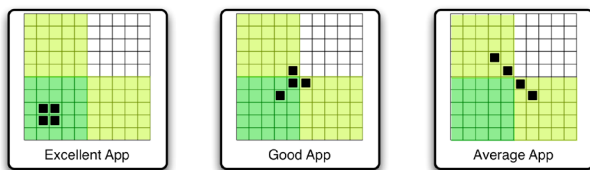


Figure 4. Excellent app, good app, and average app.

We model *changes in app infectiousness* by performing the experiments for both infectious and non-infectious apps (i.e., apps that users tell all their friends about, and apps that nobody tells their friends about).

Results and Analysis

Table 2 summarises the results of the experiments. We analyse the results in terms of each causal factor in turn.

Host Exposure. For the good app and average app, NAC is the most effective publicity strategy. For example, as can be

seen in Table 2, the average app received about 170 downloads when NAC is used, but approximately 5 downloads or less when other publicity strategies are used, and 0.26 downloads with No Exposure. Appearing on the new apps chart gives the app a high visibility over a number of weeks and users who access the app store will be able to see the app. The second most effective publicity for the good and average app is Mass Exposure. Mass Exposure reaches many users, which increases the app's chances of reaching susceptible users.

This result is the same for excellent, but non-infectious app. But surprisingly, when the excellent app is infectious, No Exposure produced the highest average number of downloads compared to the other publicity strategies. However, the standard deviation for excellent infectious apps is very large, which means that the number of downloads vary greatly in different runs. As a result, none of the publicity strategies is a clear winner or loser. Investigation of individual runs reveals that when the app is highly infectious, increasing the app's exposure to users increases the number of infected users very quickly, then the number of susceptible users falls below the level required to sustain transmission for future epidemics (everyone is already infected and is thus immune). As a result, the app falls out of the Top Apps Chart, and is forgotten.

TAC is often the worse publicity strategy. Competition from other apps means that an increase of 10,000 downloads is insufficient to boost the app into the Top Apps Chart. Targeted and Recurring Exposure are comparable, with Targeted Exposure achieving slightly more downloads. Recurring Exposure increases the chances of an epidemic, while Targeted Exposure ensures that the target tells his friends about the app. Interestingly, Targeted Exposure for excellent non-infectious app has a large standard deviation of 8 times the average. This is caused by one lucky run, whereby the publicity boosted the app into the Top Apps Chart. Once there, the high visibility created more downloads for the app, which in turn, maintained its position on the chart. As a result, the app remained in the chart for 152 days and received more than 4000 downloads. All other runs have approximately 12 downloads on average.

Figure 5(left) illustrates the number of daily downloads received by the good app in one run when Mass Exposure is used. The graph exhibits the classic epidemic curve for infectious diseases (Dicker et al., 2006). Real apps show similar epidemic curves following publicity (Figure 5(right)). The curve is the same for an excellent app, but with a larger magnitude and shorter duration. Epidemics can occur more than once in an app's lifecycle, especially for excellent apps.

Host Susceptibility. The more susceptible the users are to the app, the more downloads the app receives. As can be seen in Table 2, the excellent app receives the highest number of downloads, followed by the good app and the average app. Nevertheless, high host susceptibility alone does not guarantee downloads. With hundreds of thousands of apps in the app store, an app can easily receive no downloads just because users are unaware of the app. With No Exposure, in 3 out of 100 runs, the excellent app received zero downloads. The number of runs with zero downloads increases as the host susceptibility decreases: the good app received zero downloads in 44 runs and the average app in 87 runs.

¹² <http://blog.smashapp.com/tag/new-and-noteworthy/>

¹³ <http://forums.toucharcade.com/showthread.php?t=14134>

Strategy	Infectious			Non-infectious		
	Excellent App	Good App	Average App	Excellent App	Good App	Average App
No Exposure	6201.11 (1768.24)	694.58 (707.44)	0.26 (0.81)	3.32 (1.80)	0.77 (0.89)	0.26 (0.54)
Mass Exposure	5829.48 (1681.26)	935.53 (120.93)	5.19 (4.90)	4188.15 (657.08)	13.26 (24.53)	2.88 (1.69)
Targeted Exposure	5889.71 (1721.91)	892.86 (319.30)	0.71 (1.27)	53.49 (409.15)	3.78 (1.62)	1.71 (0.71)
Recurring Exposure	5832.04 (1338.54)	913.66 (515.99)	0.71 (1.27)	6.22 (2.12)	1.51 (1.07)	0.36 (0.66)
Enhancing thru TAC	5818.77 (1847.14)	623.34 (708.93)	1.29 (0.81)	4.02 (1.88)	1.76 (0.84)	1.23 (0.49)
Enhancing thru NAC	5840.05 (1610.12)	1020.07 (67.89)	172.48 (19.50)	4258.01 (517.44)	490.36 (58.39)	123.46 (17.69)

Table 2. Total downloads averaged over 100 runs (std. deviation in brackets). One download is equivalent to 10,000 real downloads.

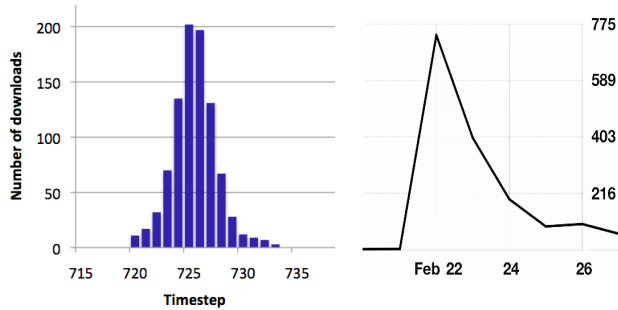


Figure 5. Left: An epidemic curve for a good app resulting from Mass Exposure in an example run. Right: Spike in app downloads as reported by Apple to the second author for his iStethoscope Pro app after a publicity event.

App Infectiousness. Infectious apps tend to receive more downloads than non-infectious ones. As can be seen in Table 2, without publicity, the excellent non-infectious app received approximately 3 downloads, but the excellent infectious app received more than 6000 downloads. Similarly, the good but non-infectious app received approximately 1 download with No Exposure, while its infectious counterpart received approximately 700 downloads. However, when apps have average features, being infectious produces a similar number of downloads as being non-infectious.

Non-infectious apps can still be downloaded at an epidemic proportion. But there are two conditions for that to happen: the host must be very susceptible to the app and the app has to be publicised, with the most effective strategy being NAC, followed by Mass Exposure and Targeted Exposure. An excellent non-infectious app receives a similar number of downloads to its infectious counterpart when either Mass Exposure or NAC is used (Table 2).

We can analyse app infection in more detail by examining infection networks for the apps. These can be categorised into three types. Type A: small network diameter, high average path length, and high average number of nodes per component (due to a few disproportionally large components); Type B: lower average number of nodes per component and all components are of similar sizes; Type C: no network. (Network diameter is the largest distance between two nodes. The diameter of a disconnected network is the maximum of all diameters of its connected components. Path length is the average graph-distance between all pairs of nodes. Connected

nodes have a graph distance of 1.) Figure 6 illustrates the Type A and Type B networks for the excellent app when Mass Exposure and NAC are used. The networks for the good app are similar but at a smaller scale.

Type A networks are produced when initial users who downloaded the app convinced their friends to download the app, who in turn, convinced their friends to download the app, and the recommendations snowball into one large connected component of the network. As the epidemic subsides, later users create isolated small clusters as most of the users in the large network are now immune to the app. Type B networks are produced when the users' friends are immune to the app because they have downloaded the app or they do not like the app. This causes snowballing to stop after a few rounds of recommendations and thus each component has a small number of nodes. Type C network occurs when the app is non-infectious, the app has poor features, or no users are aware of the app. Apps with the most downloads tend to have Type A networks. This result combined with the other results, suggests that in order to produce successful app epidemics, factors such as high user susceptibility, high app infectiousness and strategies such as enhancing through NAC and Mass Exposure are most important.

Conclusion

In this work, we used AppEco to investigate the effect of publicity on app downloads, examining which factors induce app epidemics in these complex ecosystems. We described AppEco – an Artificial Life agent-based model that simulates app ecosystems. AppEco models developers (agents that build apps), users (agents that download apps), and apps (artefacts produced by the developers and downloaded by users). It simulates the app store environment and the population growth of the agents and apps. In this work we investigated three causal factors for an app epidemic: the users' exposure to the app, their susceptibility to the app, and the infectiousness of the app.

Results show that enhancing the mode of transmission to users, specifically by having the app appear on the New Apps Chart, results in the highest chance of an epidemic occurring, and producing a massive increase in downloads. The more susceptible the users are to the app (i.e. the more users like the app), the more downloads the app receives. However, due to the massive number of apps, high susceptibility alone does not

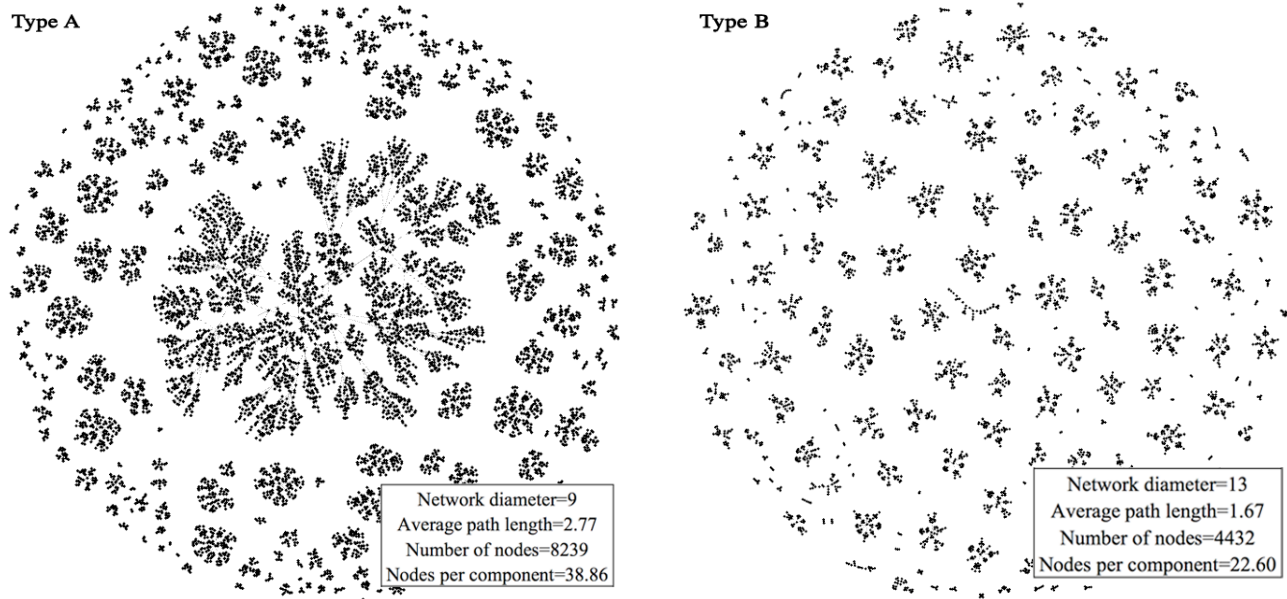


Figure 6. The spread of the excellent infectious app through the user network. Left: Mass Exposure, Right: New Apps Chart.

guarantee downloads: a highly desirable app may still receive no downloads just because users are unaware of it. Infectious apps (which encourage people to tell each other about the app) are also more likely to trigger an epidemic and receive more downloads than non-infectious apps. Non-infectious apps can still be downloaded at an epidemic proportion, but users must be very susceptible to the app and the app has to be publicised, best by the New Apps Chart strategy, followed by Mass Exposure and Targeted Exposure.

This study is one of many we will be undertaking with AppEco. For future work, we plan to investigate the effectiveness of the publicity strategies at different stages of the ecosystem, their ideal magnitude and frequency. We also plan to model app immunity caused by apps that “vaccinate” the population. Finally, AppEco can also be calibrated to study other app ecosystems, such as Android and Blackberry, and web-based platforms such as Facebook and Chrome.

References

- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6): 734-749.
- Agar, M. H., & Wilson, D. (2002). Drugmart: Heroin epidemics as complex adaptive systems. *Complexity*, 7(5): 44-52.
- Axelrod, R., & Hamilton, W. D. (1981). The evolution of cooperation. *Science*, 211(4489): 1390.
- Bohmer, M., Hecht, B., Schoning, J., Kruger, A., & Bauer, G. (2011). Falling asleep with Angry Birds, Facebook and Kindle: A large scale study on mobile application usage. *MobileHCI 2011*, pages 47-56.
- Cusumano, M. A. (2010). Platforms and services: Understanding the resurgence of Apple. *Communications of the ACM*, 53(10): 22-24.
- Dicker, R., Coronado, F., Koo, D., & Parrish, R. G. (2006). *Principles of Epidemiology in Public Health Practice* (3rd ed.). U.S. Department of Health and Human Services, Centers for Disease Control and Prevention.
- Dorin, A. (2005). A co-evolutionary epidemiological model for artificial life and death. *8th European Conference on Advances in Artificial Life (ECAL'05)*, pages 775-784.
- Dunbar, R. I. M. (1992). Neocortex size as a constraint on group size in primates. *Journal of Human Evolution*, 22(6): 469-493.
- Garg, R., & Telang, R. (2011). *Estimating App Demand from Publicly Available Data*. School of Information Systems and Management, Heinz College, Carnegie Mellon University.
- Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems* (2nd ed.). MIT Press, Cambridge, MA.
- Jansen, S., Finkelstein, A., & Brinkkemper, S. (2009). A sense of community: A research agenda for software ecosystems. *31st International Conference on Software Engineering, Companion Volume*, pages 187-190.
- Kingsland, S. E. (1995). *Modeling Nature: Episodes in the History of Population Ecology*. University of Chicago Press.
- Lim, S. L., & Bentley, P. J. (2012). How to be a successful app developer? Lessons from the simulation of an app ecosystem. *Genetic and Evolutionary Computation Conference (GECCO'12)*, in press.
- Lin, F., & Ye, W. (2009). Operating system battle in the ecosystem of smartphone industry. *International Symposium on Information Engineering and Electronic Commerce*, pages 617-621.
- Maynard Smith, J., & Price, G. R. (1973). The logic of animal conflict. *Nature*, 246(5427): 15-18.
- Olague, G., Fernandez, F., Perez, C. B., & Lutton, E. (2006). The infection algorithm: An artificial epidemic approach for dense stereo correspondence. *Artificial Life*, 12(4): 593-615.
- Rothman, K. J., Greenland, S., & Lash, T. L. (2008). *Modern Epidemiology* (3rd ed.). Lippincott Williams and Wilkins.

An Ant-Based Computer Simulator

Loizos Michael and Anastasios Yiannakides

Open University of Cyprus

loizos@ouc.ac.cy and anastasios.yiannakides@st.ouc.ac.cy

Abstract

Collaboration in nature is often illustrated through the collective behavior of ants. Although entomological studies have shown that certain goals are well-served by this collective behavior, the extent of what such a collaboration can achieve is not immediately clear. We extend past work that has argued that ants are, in principle, able to collectively compute logical circuits, and illustrate through a simulation that indeed such computations can be carried out meaningfully and robustly in situations that involve complex interactions between the ants.

Introduction

Among the central questions in studying the collective workings of agents is to understand the ways in which their interaction can produce behavior that surpasses that of the individuals. Nature offers many prototypical illustrations of this phenomenon: the flocking behavior of birds, the exploration of ants, the synchronized contraction of heart cells, all give an agent-collective the ability to perform some tasks that its individual members are unable to carry out by themselves.

We focus in this work on the collective behavior of ants. That ants work together to achieve certain common goals is, perhaps, unquestioned. It is well known, for instance, that ants can find a short route between their nest and a source of food (Deneubourg et al., 1990), sort the food, their young, and their dead into different piles (Deneubourg et al., 1991), or be recruited for some task when pheromone concentration exceeds some threshold (Bonabeau et al., 1998). What is admittedly harder to pin-point is the extent of the goals that are, *in principle*, achievable by the modus operandi of ants.

Previous work has investigated this exact question, showing that ant-like behavior is capable of universal computation (Michael, 2009), by proposing a biologically and physically plausible model for ants and pheromone, and establishing — through analysis and experimental results — its sufficiency both for the design of the basic components found in modern digital computers, and for the simulation of a logical inverter, the component that lies at the heart of logic circuits.

This work pushes that earlier investigation to its natural conclusion, by showing that the ideas presented therein can

be applied to the simulation of full-fledged circuits. In the process of establishing this claim, we are led to identify and clarify aspects of the original model, and answer questions that become important only in complex multi-gate settings.

To the best of our knowledge, this work is the first attempt to establish the *principled* capabilities of collective ant-like behavior in such a setting. This is not to suggest that others have not investigated related questions. Ant-Based Clustering (Lumer and Faieta, 1994) and Ant Colony Optimization (Dorigo and Stützle, 2004) techniques are, for instance, inspired by the behavior of ants, and are employed at large-scale settings with often remarkable results. Unlike those lines of research, our aim is not to develop ant-inspired techniques that solve particular real-world problems, but rather to investigate the capabilities of ant-like behavior itself.

Perhaps closer in spirit to our investigation are attempts to simulate circuits using nature-inspired substrates, such as proteins within living cells (Knight and Sussman, 1998), or fluids running through narrow corridors (Vestad et al., 2004). Unlike the focus of those works on very simple circuits (necessitated, respectively, by the need to employ different proteins or fluid colors across gates), this work seeks to show that circuits with no predetermined number of gates can be meaningfully considered and simulated, and explicitly sets to address the problems that arise in such complex settings.

The design of multi-agent simulators has also been investigated before (see, e.g., (Minar et al., 1996; Luke et al., 2005; Michael et al., 2010) and references therein). Unlike the generality and the generic visualization of those simulators, our aim here is to demonstrate a particular aspect of the behavior of ants, and to visualize them in a manner that would highlight its specifics. The sufficiency of using other general multi-agent environment simulators for our purposes remains an interesting question for future study.

Ant-Based Computing Basics

We start by briefly reviewing in this section the model of ant-based computing that we employ herein (Michael, 2009).

According to this model, then, the environment evolves in discrete time steps. During each time step t , any given

CHOOSEACTION(Current Location L_C , Current Direction D_C)

- 1: Identify the set $\mathcal{R}(L_C, D_C)$ of locations reachable in one step.
- 2: Sense pheromone $\mathcal{P}(L)$ for each $L \in \{L_C\} \cup \mathcal{R}(L_C, D_C)$.
- 3: If $\mathcal{P}(L_C) \geq T + \varepsilon$ and there exists $L \in \mathcal{R}(L_C, D_C)$ s.t. $\mathcal{P}(L) \leq T - \varepsilon$, then secrete s units of pheromone at L_C .
- 4: Choose $L_N \in \mathcal{R}(L_C, D_C)$ w.p. $\mathcal{P}(L_N)^n / \sum_{L \in \mathcal{R}(L_C, D_C)} \mathcal{P}(L)^n$.
- 5: Move to location L_N with direction D_N defined by $\overline{L_C L_N}$.

Figure 1: Algorithm for the behavior of each individual ant.

ant behaves as in Figure 1: The ant senses the pheromone in locations $\mathcal{R}(L_C, D_C)$ reachable from its current location L_C and direction D_C , and selects to move to a new location among the reachable ones, with the new location L_N chosen in a non-linear probabilistic fashion (determined by parameter n) according to its pheromone concentration $\mathcal{P}(L_N)$. Before moving, the ant secretes s units of pheromone at its current location L_C if it so happens that the concentration of pheromone $\mathcal{P}(L_C)$ at L_C exceeds (by a margin of ε) some threshold T , whereas this is not the case (by a margin of $-\varepsilon$) for some reachable location. In doing so, the ant helps propagate the high pheromone concentration that it has sensed.

In addition to ant movements, the state of the environment evolves also due to changes in pheromone concentrations:

$$\mathcal{P}^t(L) = (1-d) [(1-f)\mathcal{P}^{t-1}(L) + f\mathcal{W}^{t-1}(L)] + s^{t-1}(L) + p(L)$$

The pheromone concentration $\mathcal{P}^t(L)$ at location L and time-point t is determined by the concentration $\mathcal{P}^{t-1}(L)$ at that location and the average concentrations $\mathcal{W}^{t-1}(L)$ at adjacent locations at the preceding time-point $t-1$ (with diffusion between locations determined by some rate f), by its dissipation into the environment with some rate d , and by its increase by ant secretions $s^{t-1}(L)$ (as dictated by the algorithm in Figure 1) and pumps with time-invariant rate $p(L)$.

Based on this model, an inverter design was proposed:

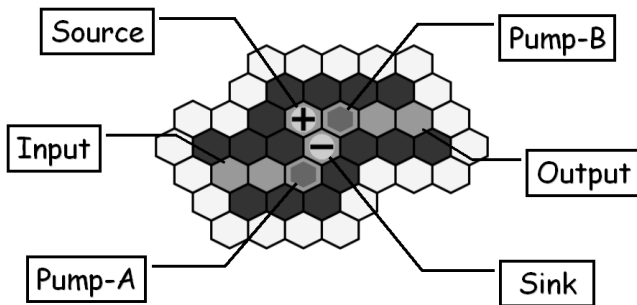


Figure 2: An ant-based inverter.

Ants are introduced in the inverter at the Source cell. The pheromone concentration at the Pump-B cell is such that it sufficiently exceeds that at the Sink cell, leading these ants

towards the Output cell. Therefore, when ants pass the Input cell at sufficiently low rate, ants pass the Output cell at sufficiently high rate. On the other hand, when ants enter at the Input cell, they head towards the Sink cell where they exit. The pheromone concentration at cell Pump-A is such that arriving ants are made to secrete additional pheromone. This increases the pheromone concentration at the Sink cell sufficiently, so it exceeds that at the Pump-B cell, leading the ants that are introduced at the Source cell towards the Sink cell. Therefore, when ants pass the Input cell at sufficiently high rate, ants pass the Output cell at sufficiently low rate.

Simulation results of a single inverter in previous work have shown that it is possible to choose appropriate values for the various parameters of the model so that the inverter behaves as expected. Additional components were proposed and had been used to design larger circuits, although no simulation results were given for them. We direct the reader to the work that introduced the model for more details.

Simulating Full-Fledged Circuits

Among the directions for future work in (Michael, 2009), and one of the central goals of this work, is the development of a tool for designing and simulating full-fledged ant-based circuits. Figure 3 presents screen-shots of the resulting tool.

We shall not go into details in describing the tool, other than briefly noting some of its main features. *Design:* Large design surface with the ability to zoom in and out and a mini-map showing the visible area, with the ability to easily place components, copy and paste parts of circuits and save them on disk for later use, and easy selection of the model parameters. *Simulation:* The simulation speed can be specified (including a single-step mode), cells can be probed and the presence of ants and pheromone can be plotted in real time, and snap-shots of the circuit state can be easily created.

We note that the tool's interface and engine offer certain functionality that is not explicitly mentioned in the original model. Indeed, in the process of developing the tool, it was clear that parts of the original model had to be made more precise, since although they sufficed for the simulation of a single inverter, they did not suffice for full-fledged circuits.

The first model extension was a clear specification of what constitutes a reachable location for an ant at an arbitrary circuit position: an ant can reach any of the three (out of all six) adjacent locations that lie directly or diagonally in front of the ant's current location and given its current direction, and only those among these three that are designated as paths. At the same time, this necessitated a clear treatment of the direction of ants, which now has to be explicitly represented and reasoned with. Accordingly, the design tool allows the placement of ants facing in any of the six possible directions.

Given this natural choice on the movement of ants, we found that the assumption of the original model that ants coming into a merge point necessarily exit from the outgoing path (and never from the other incoming path) was not read-

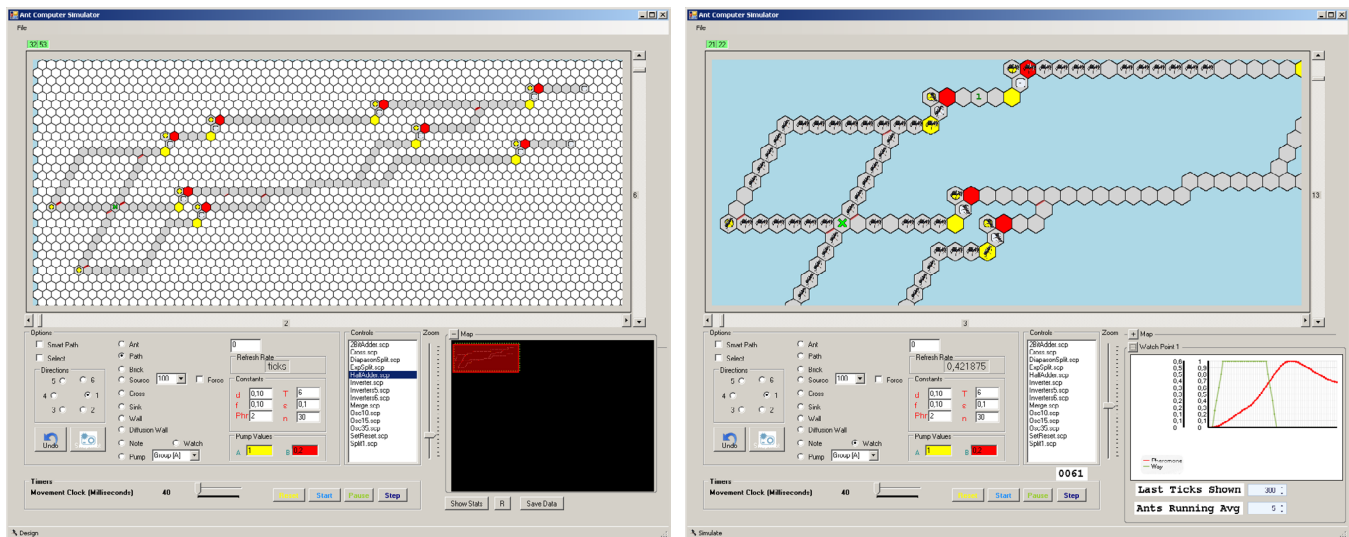


Figure 3: Screen-shots of the design (left) and simulation (right) interfaces of the developed tool.

ily realizable. This necessitated the introduction of inter-cell walls to prevent ants from crossing from one cell to another. Such inter-cell walls can be placed on any of the six sides of a cell, and can even work as one-way walls (although this was not found to be necessary for the building of circuits).

Although a bridge component (i.e., a special cell that allows two paths to cross without the two ant flows mixing) was proposed and used in the original model, an explicit implementation was not defined therein. In the present work the bridge components were implemented, allowing, even, more flexibility with three paths crossing simultaneously.

The importance of all the preceding extensions not withstanding, by far the most important extension of the original model, which necessitated both conceptual and certain engineering effort, was the proper treatment of collisions between ants. The original algorithm for the behavior of ants, as presented in Figure 1, treats each ant independently, ignoring the possibility of collisions between ants, as this was already sufficient for the simulation of a single inverter.¹

Multiple approaches for tackling this problem were considered, including allowing multiple ants to occupy the same location (i.e., making paths sufficiently wide), or making all but one ant “disappear” when multiple ants collide and claim the same location. Both solutions were found to have drawbacks that led to their dismissal: The first one would open up the possibility for ants to turn around and start moving backwards in a path, which would not be in line with the assumptions of the original model (Michael, 2009) or of experiments with real ants (Deneubourg et al., 1990). The second one is clearly physically flawed, which would compromise

¹Ants collide within an inverter *only* when they attempt to reach its sink. Exactly because the collision cell is a sink, ants that claim that cell simultaneously can be simply removed from the system without worrying about any cascade effect to the ants behind them.

the biological and physical plausibility that was claimed for the particular model that we consider and employ herein.

A third, and the most natural, choice would be to disallow an ant from moving to a location if at the time of its decision making the location is occupied by some other ant. Unfortunately, due to the discretization of time and the sequential consideration of ants, this solution would cause ants to keep away from preceding ants, leading to the formation of gaps in their flows. We considered this choice to be suboptimal by analogy to cars in a slowly moving traffic: cars flow without large gaps, under the assumption that a car currently occupying the position that another car wishes to claim will itself move away by the time the latter car reaches that position; in those cases that a car fails to move, a trickling effect of cars breaking ensues, causing cars further back in the line to remain still. Based on this analogy we have sought to find an algorithm that would produce a similar-looking behavior.

The solution we ended up adopting and implementing, illustrated in Figure 4, effectively amounts to carefully choosing the order in which ants are considered and moved. Each ant behaves as in the original model (cf. Figure 1), except that instead of moving from location L_C to location L_N , it simply claims L_N , resulting in L_C being included in $\mathcal{C}(L_N)$. A list \mathcal{E} is maintained, comprising (at every single instance within the time step being considered) locations that are simultaneously empty and claimed, so that ants claiming such locations are allowed to move first. An ant’s movement from L_C to L_N prevents other ants from moving to L_N , by having location L_N removed from list \mathcal{E} . At the same time, the movement frees up location L_C , which, if claimed by an ant, is included in list \mathcal{E} to ensure that it is eventually occupied.

Theorem 1 (Termination, Safety, Liveness) *Consider any given circuit and any model parameters. Termination: Each*

COLLISIONHANDLING(Current Environment State S_C)
1: Execute steps 1–4 of algorithm ChooseAction(\cdot, \cdot) for each ant.
2: Let $\mathcal{C}(L)$ include the locations of all ants claiming location L .
3: Set the new state S_N to be the same as the current state S_C .
4: Populate list \mathcal{E} with every location L such that $\mathcal{C}(L) \neq \emptyset$, and such that L is not occupied by any ant in the new state S_N .
5: While \mathcal{E} is not empty, remove L_N from \mathcal{E} , and do:
6: Choose at random a location L_C from $\mathcal{C}(L_N)$.
7: Move in S_N the ant from location L_C to location L_N .
8: If $\mathcal{C}(L_C) \neq \emptyset$, then include location L_C in \mathcal{E} .

Figure 4: Algorithm for avoiding collisions between ants — without the part dealing with the case of bridge components.

execution of the algorithm in Figure 4 eventually terminates. Safety: After each execution, every ant will either move to its claimed location, or will stay at its current location, and no ant will move to a location occupied by another ant. Liveness: After each execution, every location that is claimed by at least one ant, will end up being occupied by an ant.

Proof: All claims follow directly from the algorithm. \square

Given the proper movement of ants, we then considered whether the interaction of ants with pheromone might conceivably lead to a pheromone flood, that would disable the system if a circuit were to be left for a long period in a particular state (say, where ants were secreting pheromone). Such concerns can be dismissed due to the following result:

Theorem 2 (Upper-Bounded Pheromone Concentration)

Consider any given circuit and any model parameters such that $d > 0$. Let p_{max} be the maximum rate at which a pump introduces pheromone. Then, for every location L and every time-point t , it holds that $\mathcal{P}^t(L) \leq (s + p_{max})/d$.

Proof: By induction it holds that both $\mathcal{P}^t(L)$ and $\mathcal{W}^t(L)$ are at most equal to $(s + p_{max}) \sum_{i=0}^{t-1} (1-d)^i$ for every location L . The claim follows by the sum of the geometric series. \square

A last issue considered was the hysteresis of circuits: the time that elapses between setting the circuit inputs until the circuit output computes the correct result. Indeed, even if one knows what output a circuit is supposed to produce, just observing that the output is correct at some time-point is not an indication that it will not fluctuate in the future. A more proper treatment is to wait for time at least equal to the hysteresis of the circuit before observing the output.

Definition 2.1 (Acyclic Circuit Hysteresis) *Consider any given acyclic² circuit and any model parameters. For each input-to-output path in the circuit, count the number k of inverters, and the number m of cells that appear in paths outside the inverters. Then, the hysteresis of that path is*

²For cyclic circuits the definition of hysteresis is unclear, since it is possible that a cyclic circuit might not stabilize eventually.

$hk + m$, where h is the hysteresis of a single inverter (the only computational unit in a circuit). The hysteresis of the entire circuit is the maximum hysteresis of its paths.

The hysteresis h of an inverter was obtained empirically, and the employed empirical setting will be presented later.

Experimental Setting and Results

The development of the simulator just described provides the necessary means to pursue the second main goal of this work: the empirical verification that the model described in the previous sections (along with the proposed modifications) works as expected, and scales up to full-fledged circuits. We present in this section a comprehensive series of experiments and empirical results towards this goal.

Through the developed tool, we have designed various circuits, or parts thereof, and have placed probes on certain cells. Each probe is numbered, and measures at each time-point both the pheromone concentration at that location, and the presence or absence of an ant. Each of the figures that follows presents the probed circuit along with the locations of its probes, followed by graphs plotting pheromone concentration or (the running average of) ant presence at the locations of each of these probes. We discuss certain specifics of each experimental setting when we present it below.

In the first experiment we investigate one of the most immediate concerns when simulating complex circuits: the fact that the diffusion of pheromone from one inverter could potentially reach, and interfere with the operation of, another inverter. This interference is not unlike what happens in actual electronic circuits when components are close together.

Figure 5 shows a single path equipped with a pheromone pump, whose rate is chosen to equal the highest among those used for the pumps of an inverter. The two graphs present the same information in two different ways. Each line in the first graph presents the pheromone concentration for each of the probed locations, at a particular time-point. In combination with the second graph, it is clear that pheromone concentrations are stable by time-point 100. Hence, no matter how much time one waits, cells that are at distance 4 and above from the pump have essentially zero pheromone concentration. By choice of the pump rate, and assuming that inverters are as presented in Figure 2, perhaps with slightly longer Input and Output paths, we can ensure the non-interference between inverters. Of course, if inverters that employ pumps with other rates were to be used, then longer paths would be needed. In all cases, the inverter designer can include as part of the inverter sufficiently long paths to achieve the needed insulation of the inverter from outside interference.

Other important conclusions also follow from the first experiment. First, the pheromone concentrations do not increase unboundedly, as already established more generally by Theorem 2. Second, the first graph shows that the model of pheromone diffusion considered is plausible, with a peak at the pump location, and rapid decay as one moves away.

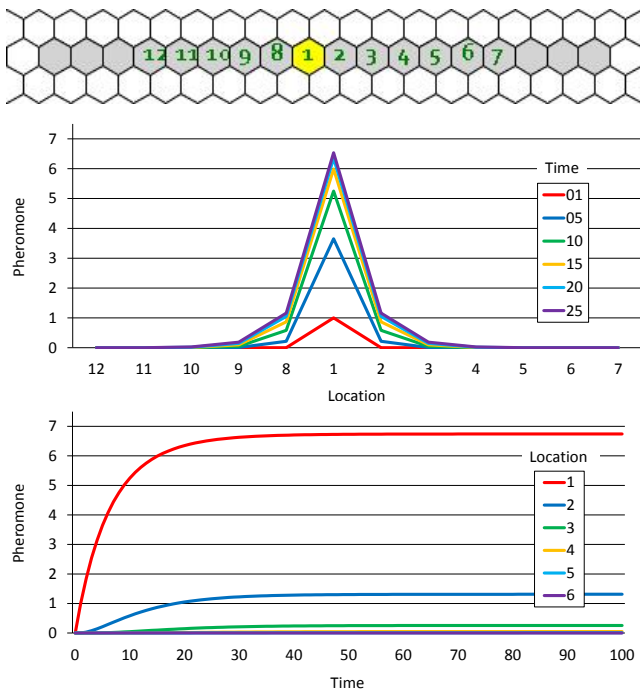


Figure 5: Pheromone concentrations over time and distance from a pump. Even after pheromone concentrations stabilize, the pheromone diffusion is bounded close to the pump.

In the second experiment we confirm the characteristic of ants to follow high pheromone concentrations (cf. Figure 1).

Figure 6 shows a choice point for ants, with each outgoing path containing a pump. The (running average) presence of ants at the two outgoing paths is plotted over time. In the first graph both pheromone pumps are deactivated, and ants are shown to uniformly at random select an outgoing path. One may also note that the sum of outgoing ants is always steady, providing some empirical verification to Theorem 1.

In the second graph both pumps are activated, with the red pump having a fixed rate of 0.100, and the yellow pump's rate ranging from 0.100 to 0.113 across different runs. Only the ants choosing the upper path are plotted over time, for each run. As expected, the higher the difference between the two pump rates, the higher the difference of concentrations at the two outgoing cells of the choice point, and the more likely an ant is to choose to follow the upper path. Our choice of model parameters for the experiments presented herein shows that ants are rather sensitive to small changes of pheromone concentrations. As it follows from the second graph, a 15% increase in pheromone is sufficient to attract all ants towards the higher concentration. We emphasize that this sensitivity is due to the particular model parameters that we have used, and that other choices of the model parameters would have produced less or more sensitive ants.

The reader may note that the behavior of ants is mostly time-invariant. Indeed, for this experiment we have purpose-

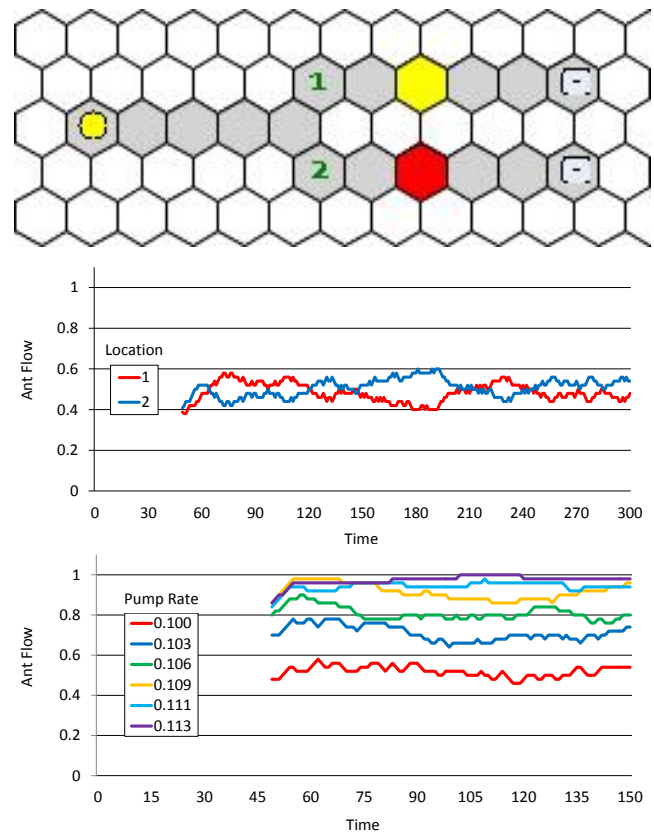


Figure 6: Ants choosing a path based on pheromone concentrations at the choice point. The more distant the two pheromone concentrations, the more steeply ants choose to move towards the higher pheromone concentration.

fully chosen the pump rates to be sufficiently low, so that the pheromone concentrations never increase beyond what would cause ants to secrete even more pheromone, and thus, tilt the choice of all subsequent ants towards one direction.

The next natural point of investigation is whether ants will indeed affect each other in the presence of sufficiently high pheromone concentration. The third experiment illustrates that ants that reach the pump secrete additional pheromone, which eventually reaches the choice point and affects the behavior of other ants. Thus, the precise phenomenon that we sought to avoid earlier is what is investigated next.

Figure 7 shows a setting rather similar to the previous experiment. The rates of pumps are now those found in an inverter. Thus, if these two pumps are left to operate for some time, the pheromone concentration around the yellow pump will be sufficiently high to cause ants to be recruited and secrete more pheromone. However, due to diffusion and the distance of the pumps to the choice point, the pheromone concentration at the choice point is not as much high. In fact, it is the case here that the lower exit point of the choice point has more pheromone than the upper exit point, and hence that ants are more likely to follow the lower path. But

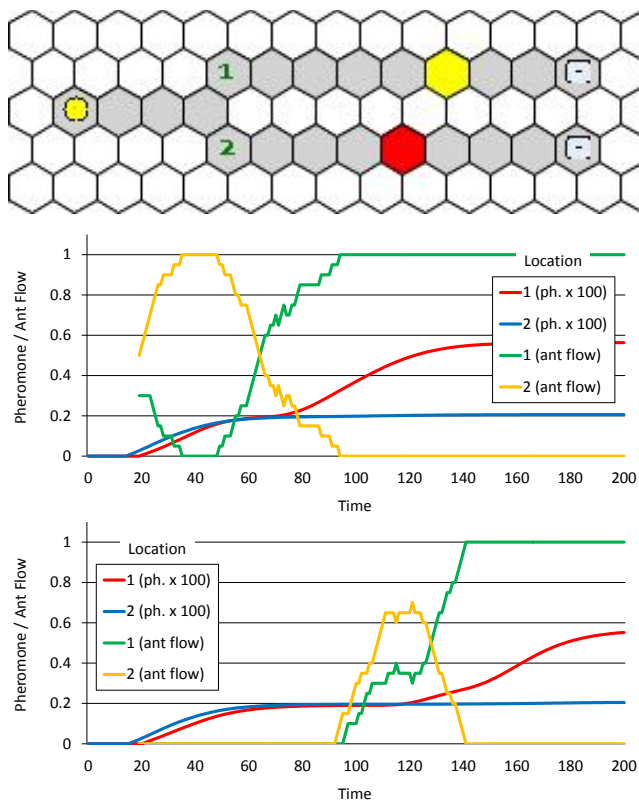


Figure 7: Ants choosing the upper path, reaching a point of sufficiently high pheromone concentration and secreting additional pheromone, leading to a phase shift in ant behavior.

once the (minority of) ants that choose the upper path reach the yellow pump, they start secreting more pheromone. The extra pheromone diffuses back to the upper exit point of the choice point, making it more likely for ants to choose the upper path, more ants reaching the yellow pump, and so on, leading to a phase shift. The two graphs present exactly this phase shift occurring, with the rapid increase in pheromone concentration. In the first graph ants were introduced from time-point 0 into the system, while in the second graph ants were introduced after pheromone concentrations stabilized.

Returning to the point of testing the working of the employed collision handling algorithm, we investigate a setting where ants collide in merged paths. Although our collision handling algorithm (cf. Figure 4) is fully general and works for any configuration (cf. Theorem 1), we present a simple to understand and quantify experimental setting for its testing.

Figure 8 shows a merging point for ants. Through the use of multiple probes, we are able to count the remaining ants in each incoming path for each time-point, and we plot their number over time, as well as the number of outgoing ants over time. The graph shows that the outgoing path consistently contains ants as long as either incoming path has ants, and then stops containing ants. This, in particular, shows that our collision handling algorithm does not suffer from

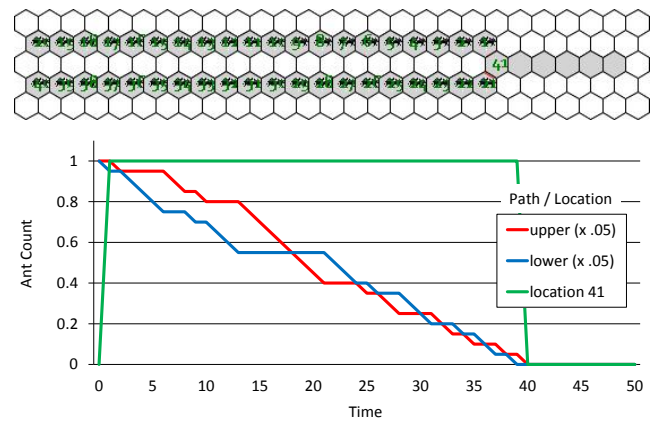


Figure 8: Ants merge into a single path, while avoiding collisions. All ants from the two incoming paths eventually reach the outgoing path, and the incoming paths are given equal preference as sources of ants. No gaps between ants are observed due to the natural handling of collisions.

typical gap-forming problems that arise from the discrete-time movement of agents. At the same time, the graph illustrates that no ants “disappear” in the process, and that equal preference is given to ants from either incoming direction.

The four experiments described above establish that the basic functionality of the simulator works as expected. The obvious next experiment, then, is that of reproducing what the earlier work had done: to simulate a single inverter.

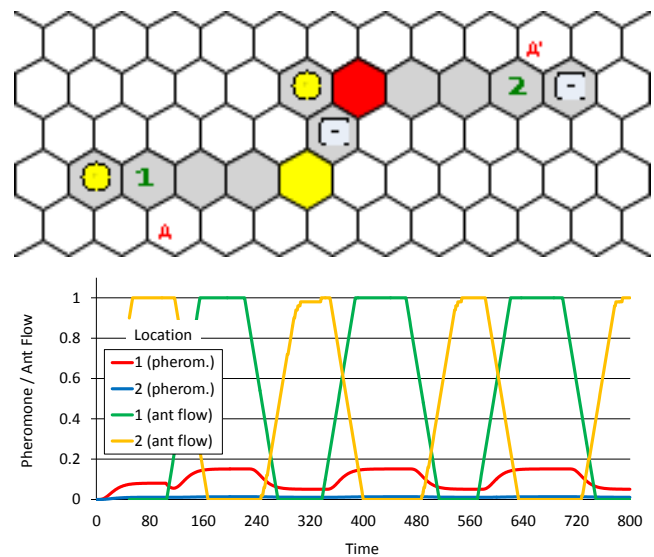


Figure 9: An ant-based inverter, inverting its input.

Figure 9 shows a single inverter implemented with the developed tool. A point at its input and a point at its output are probed. Pheromone concentrations at these two points are those that diffuse from the pumps of the inverter. Ants at the input of the inverter are controlled by the experimenter,

by opening and closing the switch near the first probe. The graph shows that the output is affected inversely. The flattening out of pheromone concentrations indicates that no matter how much time the inverter spends in any one state, changing to the other state is performed in a fixed bounded time.

This experiment could provide an indication for the hysteresis h of an inverter, but this critically depends on how one interprets the presence of an output signal (e.g., when even a single ant appears at the output, or when the ant flow is uninterrupted for some time). To obtain a less ambiguous upper bound on h , but also to investigate other aspects of complex circuits, we consider next a sequence of inverters.

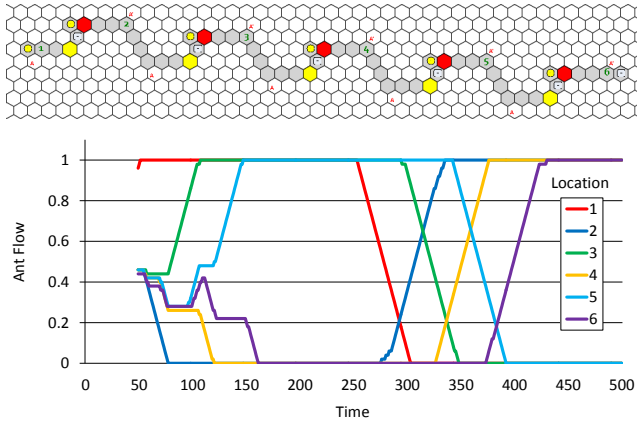


Figure 10: A sequence of inverters that stabilize over time, producing a cascade effect. Comparing the input of one inverter to the output of its successor inverter is a clean way to identify an upper bound on the hysteresis of an inverter.

Figure 10 shows a sequence of inverters, with each one's output feeding into the input of the next one. The input of the first inverter and the output of all inverters (and hence their inputs) are probed. The former is controlled by the experimenter. The graph shows the output of the inverters, where once the input is set to 1, the inverters successively, in a cascade style, flip their outputs in their given order. Once the input is set back to 0, again the cascade effect is observed.

Note that this experimental setting might be more appropriate to measure the hysteresis of an inverter. Instead of comparing a single inverter's input and output, where it is not immediately clear at which point a signal changes from 0 to 1 or from 1 to 0, one may compare an inverter's input to the output of the *next* inverter. As shown in the graph, the two signals change in parallel, with a time shift of about 50 time units. The distance between the two accounts for the change of the state of two inverters, one from 0 to 1 and one from 1 to 0, and for the delay introduced by the path connecting the two inverters. Taking into account that the two state changes may have different hysteresis, one can still safely use 50 as an upper bound of any state change in an inverter.

As a demonstration that our developed tool scales up, both

for design and simulation, we consider next a 2-bit adder.

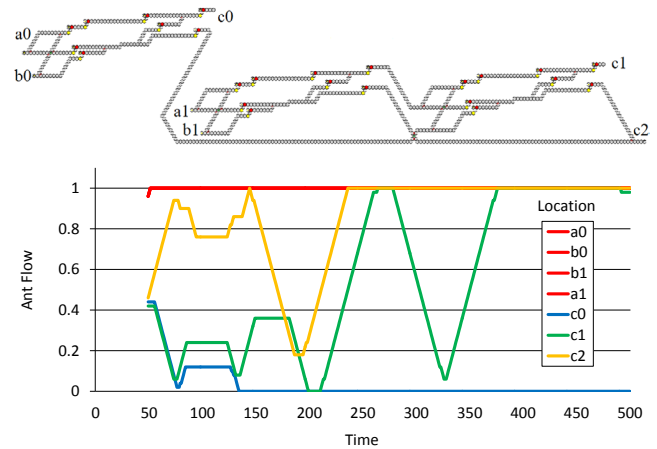


Figure 11: A two-bit adder with its inputs and the resulting outputs. Given the input 3 ($a=11$) plus 3 ($b=11$), the circuit produces the output 6 ($c=110$). During the computation, the outputs fluctuate before stabilizing to the correct values.

Figure 11 shows a 2-bit adder, with two 2-bit inputs a and b , and a 3-bit output c computing $a + b$. The graph shows the presence of ants at the circuit inputs and outputs. After about 400 time units the signals on all output wires stabilize to the correct values, after fluctuating during the computation.

Besides providing evidence that the design and simulation of large circuits is possible, this experiment offers the opportunity to validate our proposed formula for circuit hysteresis (cf. Definition 2.1). One may consider all input-to-output paths in the 2-bit adder, and observe that the longest path in terms of its hysteresis is the one from $b0$ to $c1$, with $k = 6$ inverters and approximately $m = 200$ cells in paths outside inverters. Using $h = 50$ as an upper bound for the hysteresis of an inverter, we get that a safe upper bound on the hysteresis of the circuit is $50 \cdot 6 + 200 = 500$. Indeed, the circuit computes its result before the upper bound is reached.

As a further demonstration of the functionality of our proposed simulation tool, we consider the case of a memory bit.

Figure 12 shows an SR-latch implementation of a 1-bit memory, via the use of cycles in the circuit. The figure plots the two input signals and the output signal. Initially the output fluctuates, as a result of the cyclic nature of the circuit. After some time, the set signal forces the output to become 1, which remains so even after the set signal is removed. Analogously, once the reset signal is given, the output becomes 0 and remains so even after the reset signal is removed.

Finally, we consider the implementation of an oscillator.

Figure 13 shows a single inverter whose output is fed back as its input. Depending on the length of the output-to-input path, the oscillator exhibits different oscillation frequencies. Oscillators can be used as clocks for the synchronization of other circuits, as needed to build an ant-based computer.

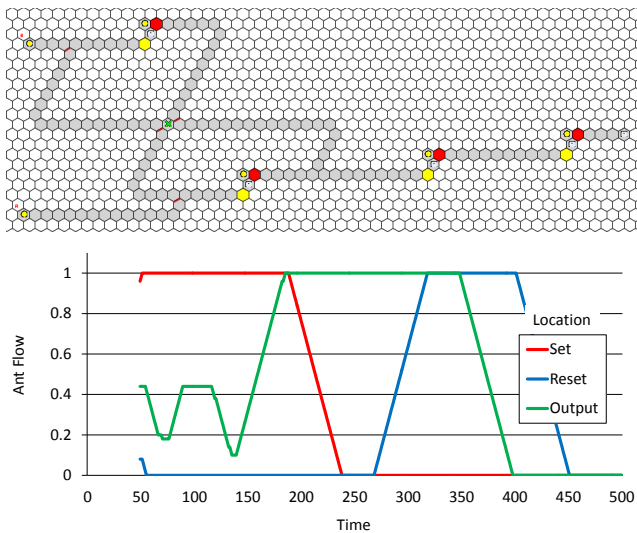


Figure 12: A single-bit memory circuit. Once it is (re)set, its output remains stable even when the input signal is removed.

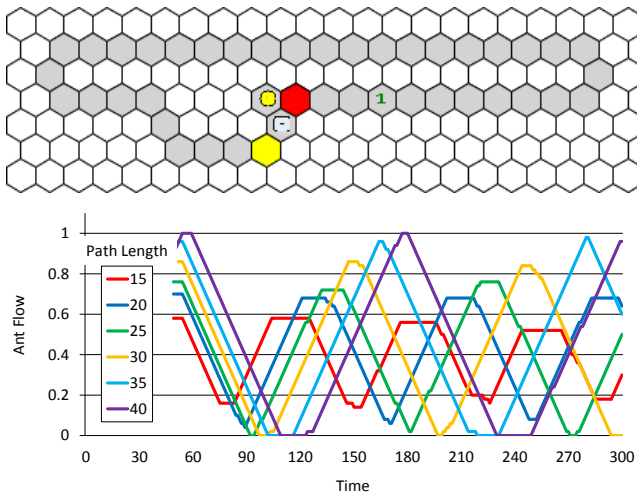


Figure 13: An oscillator with various frequencies.

Conclusions

We have presented a tool for designing and simulating ant-based computers, and have demonstrated its proper working through an extensive suit of experiments. Although one could argue that the circuits considered herein are small, or trivial, compared to those found in modern computers, the model that we have employed seems to suffer from no inherent limitations in terms of scalability to much larger circuits.

Admittedly, it is not our intention to suggest that any particular species of ants behaves in any particular manner in real life. Yet, we believe that the results in this and our earlier work (Michael, 2009) provide for the first time evidence that ants are *in principle* able to collectively behave as presented herein. We would find entomological field work attempting to corroborate this evidence especially intriguing.

Beyond its presumed biological implications, this work may also find applications in education, offering a gentle and entertaining (game-playing-like) introduction to the notions of mathematical logic, digital circuits, and computers.

In terms of future work, our directions of interest include the design and simulation of much larger circuits, and the parallelization of the collision handling algorithm. Beyond the obvious gain in speed that such a parallelization is expected to offer, there is also the more conceptual benefit of bringing the simulator closer to the distributed and asynchronous behavior of real ants, when they go marching on!

References

- Bonabeau, E., Theraulaz, G., and Deneubourg, J.-L. (1998). Fixed Response Thresholds and the Regulation of Division of Labour in Insect Societies. *Bulletin of Mathematical Biology*, 60(4):753–807.
- Deneubourg, J.-L., Aron, S., Goss, S., and Pasteels, J. M. (1990). The Self-Organizing Exploratory Pattern of the Argentine Ant. *Journal of Insect Behavior*, 3(2):159–168.
- Deneubourg, J.-L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., and Chrétien, L. (1991). The Dynamics of Collective Sorting: Robot-Like Ants and Ant-Like Robots. In *Proceedings of the First International Conference on Simulation of Adaptive Behavior: From Animals to Animats*.
- Dorigo, M. and Stützle, T. (2004). *Ant Colony Optimization*. MIT Press, Cambridge, MA, U.S.A.
- Knight, T. F. and Sussman, G. J. (1998). Cellular Gate Technology. In *Proceedings of the First International Conference on Unconventional Models of Computation*.
- Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., and Balan, G. (2005). MASON: A Multi-Agent Simulation Environment. *Simulation: Transactions of the society for Modeling and Simulation International*, 81(7):517–527.
- Lumer, E. and Faieta, B. (1994). Diversity and Adaption in Populations of Clustering Ants. In *Proceedings of the Third International Conference on Simulation of Adaptive Behaviour: From Animals to Animats*.
- Michael, L. (2009). Ant-Based Computing. *Artificial Life*, 15(3):337–349.
- Michael, L., Parkes, D. C., and Pfeffer, A. (2010). Specifying and Monitoring Economic Environments using Rights and Obligations. *Autonomous Agents and Multi-Agent Systems*, 20(2):158–197.
- Minar, N., Burkhart, R., Langton, C., and Askenazi, M. (1996). The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations. Technical Report Working Paper 96-06-042, Santa Fe Institute, Santa Fe, NM, U.S.A.
- Vestad, T., Marr, D. W. M., and Munakata, T. (2004). Flow Resistance for Microfluidic Logic Operations. *Applied Physics Letters*, 84(25):5074–5075.

Effects of public good properties on the evolution of cooperation

Dusan Misevic^{1*}, Antoine Frénoy¹, David P. Parsons², and François Taddei¹

¹Center for Research and Interdisciplinarity, INSERM U1001, University Paris Descartes, Sorbonne Paris Cité, France

²Université de Lyon, CNRS, INRIA, INSA-Lyon, LIRIS, UMR5205, F-69621, France

*corresponding author: dule@alife.org

Abstract

Cooperation is a still unsolved and ever-controversial topic in evolutionary biology. Why do organisms engage in activities with long-term communal benefits but short-term individual cost? A general answer remains elusive, suggesting many important factors must still be examined and better understood. Here we study cooperation based on the secretion of a public good molecule using *Aevol*, a digital platform inspired by microbial cooperation systems. Specifically, we focus on the environmental and physical properties of the public good itself, its mobility, durability, and cost. The intensity of cooperation that evolves in our digital populations, as measured by the amount of the public good molecule organisms secrete, strongly depends on the properties of such a molecule. Specifically, and somewhat counter intuitively, digital organisms evolve to secrete more when public good degrades or diffuses quickly. The evolution of secretion also depends on the interactions between the population structure and public good properties, not just their individual values. Environmental factors affecting population diversity have been extensively studied in the past, but here we show that physical aspects of the cooperation mechanism itself may be equally if not more important. Given the wide range of substrates and environments that support microbial cooperation in nature, our results highlight the need for careful consideration of public good properties when studying the evolution of cooperation in bacterial or computational models.

Introduction and Background

In recent years, such a complex and sophisticated array of collective behaviors has been observed in microbes that it has created and motivated a rapidly growing new field, “sociomicrobiology” (Parsek and Greenberg 2005; West et al. 2006). Arguably the most interesting among such behaviors is cooperation, which is frequently observed in nature, has been extensively studied theoretically, and in some cases, even experimentally (West et al. 2006). Cooperation in microbes can affect crucial cell processes, such as reproduction (Strassmann et al. 2000; Queller et al. 2003; Fiegna and Velicer 2005), resource sharing (MacLean and Gudelj 2006), biofilm formation (Brockhurst et al. 2006), and motility (Velicer and Yu 2003). Particularly interesting are instances when cooperation is maintained by public good, such as colicin toxins (Le Gac and Doebeli 2009), heavy metal detoxification (Ellis et al. 2007), quorum sensing (Dunny et al. 2008; Czarán and Hoekstra 2009), or triggering of host

immune response via self-destruction (Ackermann et al. 2008). In all these cases, bacteria produce a public good, a molecule or a modification of the environment that is beneficial for the entire population but produced by individuals at a cost. Researchers have been especially interested in medical implications of cooperation via public good secretion, as in the case of *Pseudomonas aeruginosa* infections of cystic fibrosis patients (Paton 1996) because cooperation breakdown would decrease pathogen virulence and could be used as a treatment strategy.

The evolution and maintenance of cooperation has remained an important biological question because it appears to contradict basic principles of natural selection: organisms that help others at the cost of decreasing their own fitness should be selected against. In a mixed population, a non-producing organism will have a higher fitness than a producing one because it does not pay any of the costs associated with public good creation and secretion. The majority of the currently accepted theories for the maintenance of public good production in microbes in spite of the direct non-producer advantage are a combination of spatial assortment (e.g. environment structure, limited dispersal, viscous environment) and kin selection (Griffin et al. 2004; Diggle et al. 2007), although other explanations are possible (Brockhurst et al. 2008; Kümmerli et al. 2009b; Ross-Gillespie et al. 2009). Simply put, the public good is expected to be maintained when it is preferentially benefiting its producers and their close relatives (Fletcher and Doebeli 2009). This theory has been experimentally tested in the past but with sometimes differing conclusions (Kümmerli et al. 2008; Kümmerli et al. 2009a). There has been significantly less work addressing durability of the public good and the environmental viscosity not just in terms of movement of the individuals but also in terms of the diffusion of the public good (however, see Brown and Taddei 2007; Kümmerli and Brown 2010). In our study we identify and quantify the effects and interactions of public good properties such as the rate of diffusion and degradation on the evolutionary trajectories of cooperative properties.

Methods

To examine the effects of public good properties on the evolution of cooperation we use *Aevol*, a computer platform

that enables tracking of large populations of digital organisms over thousands of generations. *Aevol* resembles other well-established *in silico* experimental systems, such as *Avida* (Lenski et al. 1999; Misevic et al. 2004; Ofria and Wilke 2004), but its main strength is in the greater attention that is given to the genome structure and encoding (Knibbe et al. 2006; Knibbe et al. 2008; Beslon et al. 2010). It is freely available at www.aevol.fr/download and we used the default parameters unless otherwise noted. *Aevol* has been described in great detail previously (Parsons et al. 2010) and here we highlight only its main properties and new features specifically implemented for the study of cooperation.

The *Aevol* experimental system

General properties. *Aevol* individuals mutate, interact with one another, have their fitness evaluated, and are reproduced in a typical genetic algorithm fashion. An individual is represented by its circular genome, a double stranded binary string that can be hundreds of thousands of digits long. There is a complex genotype to phenotype to fitness mapping which we briefly describe here (Figure 1).

Genotype to protein: Pre-determined binary motifs act as promoter and terminator sequences and specify the transcribed regions of the genome. Within these regions, start and stop codons mark the sequences that will be translated into proteins. Each protein's sequence is interpreted as three numerical values (m , w , h), for the mean, width and height of a triangle that represents the protein's phenotypic contribution. For more details on transcription and translation in *Aevol*, see (Parsons et al. 2010).

Proteins to phenotype: The phenotype is defined as the combination of all the expressed proteins, calculated by adding together the protein triangles. In practice, an organism's phenotype is typically a jagged, piecewise-linear function on the interval $(0,1)$.

Phenotype to fitness: We define the fitness of an organism as $W = e^{-ag}$, where g is the geometric area between the organism's phenotype and the target phenotype, and a is a selection pressure constant. The typical target phenotype is the sum of several Gaussians and remains constant over the course of a single simulation. In more general terms, we consider an organism as a collection of traits. In order to represent these traits numerically in a limited number of dimensions, we simplify things by positioning all traits on the continuous trait axis. Each protein primarily affects a single trait (determined by m , the position of the triangle on the trait axis) at a specific level (determined by h , the height of the triangle), but also the neighboring traits (to the extent determined by w , the width of the triangle, representing the pleiotropy of the protein) at a lower level. It is important to note that the same protein can be encoded by different sequences and that neighboring traits are not necessarily encoded by sequences that are similar or close to one another on the genome.

Population spatial structure: A typical *Aevol* population resembles a well-mixed bacterial population, with organisms having no specific positions in space. However, under the spatial regime we implemented for this study, organisms reside on a rectangular grid with a periodic boundary condition, i.e. a torus. After their fitness is evaluated,

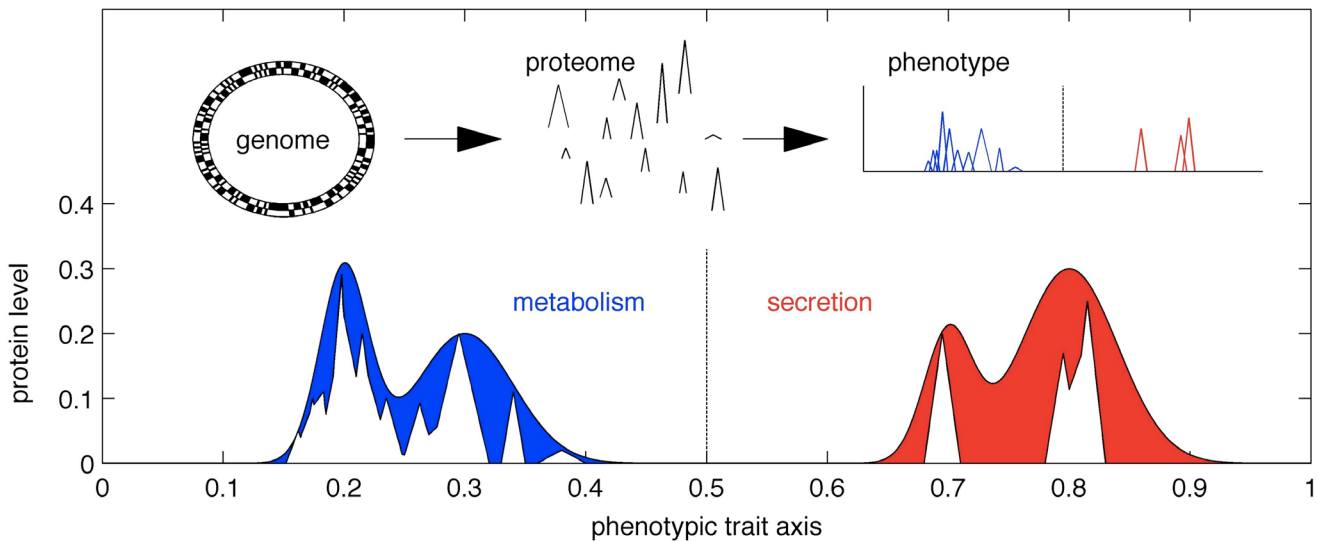


Figure 1. Genotype to phenotype to fitness mapping in *Aevol*. A circular, double stranded genome is schematically represented with black and white squares, corresponding to zeros and ones in *Aevol* (for clarity, only 100 bases are represented here, much less than the typical genome size). The transcription and translation stages produce the proteins, represented as triangles, which are located on the phenotypic trait axis. The triangles are added together to form the organism's phenotype. Two regions of the trait axis are designated for different functions, metabolism (blue) or secretion (red), and each has a separate target phenotype, here the sum of two Gaussians, as specified in the main text. The gap g (shaded region) between the phenotype and the optimal phenotype is inversely proportional to a metabolic component of fitness (blue, metabolism) or to amount of the public good that the organism secretes (red, secretion).

organisms compete with one another to produce offspring that will populate the next generation. For each position in the grid, organisms in the classical 3x3 Moore neighborhood of the position have a probability $(a - 1) \times a^{9-R} / (a^9 - 1)$ of reproducing into this position, where R is the organism's rank in the neighborhood, based on fitness, and a is a selection pressure constant. The organism that actually reproduces is then chosen using roulette selection. During replication, the genome of the new organism experiences a full range of different mutation types at user-set, per-base rates of 10^{-5} (point mutation, and small insertions/deletions of up to 6 bases) and 10^{-6} (duplications, large deletions, translocations and inversions).

In order to vary the strength of spatial structure in our experiments, at every generation we chose pairs of organisms at random and swapped their location. By increasing the migration parameter (*mig*, the number of swaps per generation), we can gradually vary the population structure from well mixed (high *mig*) to perfectly local (*mig* = 0).

Cooperation in Aevol. During their lifetime, organisms may secrete a single type of molecule that accumulates in the environment, degrades and diffuses over time, and whose uptake directly affects the organism's fitness. This molecule is the public good that enables cooperation among individuals. To enable organisms to control the level of secretion, we split the axis of phenotypic traits into two sections, metabolism and secretion (Figure 1). The metabolic traits affect fitness directly and organisms evolve to match the metabolic target phenotype. The secretion traits determine the amount of public good an organism secretes into the environment, which is reversely proportional to the gap between the organism's phenotype and the predefined secretion target function. The total fitness of an organism depends on its metabolic fitness, the cost it pays for secreting the public good, and the benefit it gets from any public good molecules already present in the environment. Specifically, the fitness is equal to $W_{met} \times (1 + PG - C \times S)$ where W_{met} is its metabolic fitness (calculated the same way as in experiments without secretion), PG is the amount of the public good molecule present in the grid cell that organisms inhabits, C is the cost of secreting a unit of public good, and S is the amount of public good molecule that the organism secretes. Additionally, $W_{met} = e^{-aG_m}$ and $PG = e^{-aG_s}$, where G_m is the gap between the target phenotype and the organisms phenotype for metabolism, G_s is the gap between the target phenotype and the organisms phenotype for secretion, and a is a selection pressure constant as before. It is important to note that an organism may not directly benefit from the molecule it secretes, as its amount is added to the environment only after the fitness of the organism is calculated. The benefit may only occur in the following generation making any selection for cooperation indirect.

Once the public good is secreted into the environment, at every generation it diffuses and degrades. We primarily think of diffusion and degradation as being dependent on the properties of the public good molecule itself (e.g. its size, hydrophobicity) but in nature they can also be affected by the environmental properties (e.g. viscosity, solubility). The diffusion is controlled by the *dif* parameter, which specifies

what percentage of the public good molecules present in a population grid location will diffuse into each of its eight neighbors in the 3x3 Moore neighborhood. Similarly, *deg* determines the percentage of the public good molecules present that is degraded at each generation, the public good durability. For example, if 2 units of public good are present, *dif* = 0.05 and *deg* = 0.2, each of the 8 neighbors will receive $2 \times 0.05 \times (1 - 0.2) = 0.08$ units of public good, while the original location will have $(2 - 8 \times 2 \times 0.05) \times (1 - 0.2) = 1.28$. Due to the multiplicative nature of diffusion, even at extremely high diffusion levels, its effects are local, less than 2% of the public good reaching grid locations at least three positions away.

To summarize, the *Aevol* system follows the genetic algorithm heuristic and each generation consists of the following steps: (1) evaluation of the organisms' fitness based on their metabolic proteins and on the amount of public good present in their local environment, (2) secretion of the public good molecule at levels determined by secretion proteins, (3) selection of the organisms that will reproduce, based on their fitness, (4) application of mutations to the new-born organisms, (5) diffusion and degradation of the public good, (6) organism migration, by swapping randomly chosen pairs of individuals. This setup enables us to study the evolution and maintenance of cooperation over thousands of generations.

Experimental design

Given the large number of parameters that could be varied in *Aevol*, it was computationally nonpermissive to examine all possible combinations of relevant public good properties. Instead, we first focused on establishing a medium level of the cost of public good secretion that enables the evolution and persistence of cooperation in our system. To do so, we performed experiments where the cost of secreting a unit of public good was 0, 0.01, 0.03, 0.1, or 0.3, while diffusion was set to 0.05 and degradation to 0.1. In a second set of experiments, we examined the effects of different levels of migration, diffusion and degradation on cooperation at a given cost level. The secretion cost was set to 0.03, while the other parameters were varied as follows: migration was 0, 100, 300, or 1000, diffusion was 0, 0.01, 0.05, or 0.1, and degradation was 0, 0.01, 0.1, or 0.3. In each set of experiments and for each combination of parameters, we evolved 10 replicate populations of 1024 individuals for 20,000 generations, starting with a different seed for the random number generator. In total, 680 experiments were run for a combined total of 13.6 million generations of evolution. All populations evolved in square toroidal grids and had the same phenotypic target function specified by four Gaussian functions of the form $y = H \exp(-(x - M)^2 / (2W^2))$, where $(H, M, W) = \{(0.2, 0.3, 0.04), (0.3, 0.2, 0.02), (0.2, 0.7, 0.02), (0.3, 0.8, 0.04)\}$. At each generation, we recorded population averages of metabolic and total fitness, amount of public good secreted by an individual, and amount of public good present in the environment. The statistical analysis was performed using Matlab R2011b.

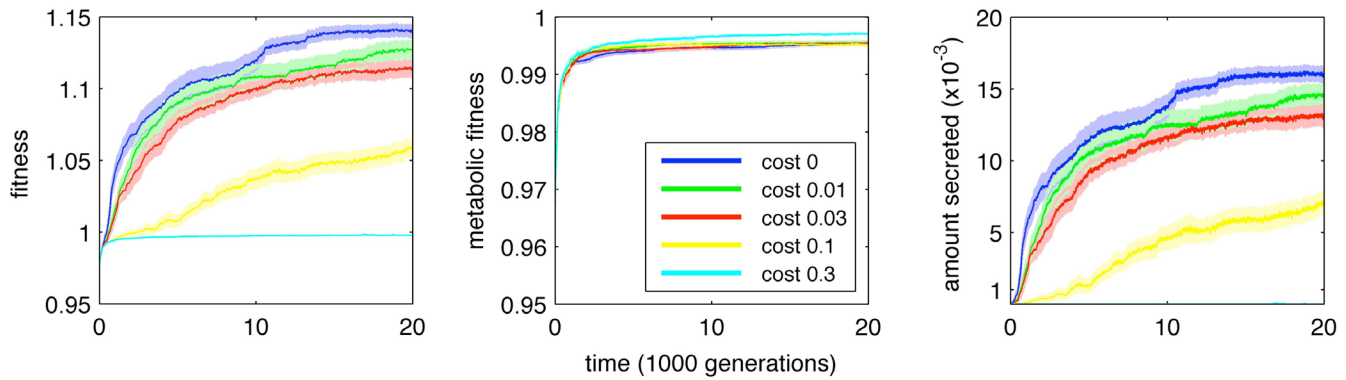


Figure 2. Trajectories for fitness, metabolic fitness and the amount of public good secreted during evolution at different costs of secretion. Each curve is the average of 10 different replicate experiments and the shaded area represents one standard error of the mean.

Results and discussion

Secretion cost and the evolution of cooperation

We find that secretion, and thus cooperation, evolves to higher levels when the cost of secreting the public good is lower (Figure 2). In and of itself, this result is not surprising. Indeed, based on the way fitness is calculated in *Aevol*, a direct tradeoff exists between the cost and benefit of cooperation. However, there are several additional observations that can be made about the results of these experiments. Unlike game-theoretical simulations of cooperation, where cooperation is a discrete trait and each organism is either a “cooperator” or

“cheater/defector”, here we have a continuum of possible cooperation levels. One could consider that any organisms secreting less than the currently maximum is a cheater. However, given the constantly changing and often increasing maximal secreted amount, we would then have to frequently relabeled cooperators as cheaters, potentially creating much confusion. Instead, we altogether avoid such binary classification and focus on the average secretion level in the population. Although the amount of secretion has not stabilized in our experiments, when we examined individual population trajectories, rather than seeing large amplitude cycles of cooperation/defection, each type taking turns in invading and (possibly) taking over the population, the dominant pattern is one of steady, stepwise increase in public good secretion, with some low level variance.

We further analyzed the dynamics of evolution by examining the diversity of phenotypes within a population and did find multiple types coexisting, but no direct evidence of cycles (Figure 3). However, based purely on diversity data we cannot discern whether independent lineages of individuals with different levels of secretion are coexisting through time or if the lower level secretion repeatedly emerges via mutations. To distinguish between these two possibilities, we performed additional experiments in which we turned off all types of mutations after 10,000 generation of evolution, switching effectively to an “ecological mode”, and recorded the amount of public good that was secreted. We analyzed 10 replicates of continued “ecological mode” for 5 different ancestral populations, such as the one in Figure 3. In all cases we obtained qualitatively identical results: within a few hundred generations, a single genotype, that with the highest secretion level, swept through the entire population (data not shown). We conclude that, at least at that stage of evolution, the diversity in the amount of public good secreted is due to a constant supply of mutations and not to ecological interactions.

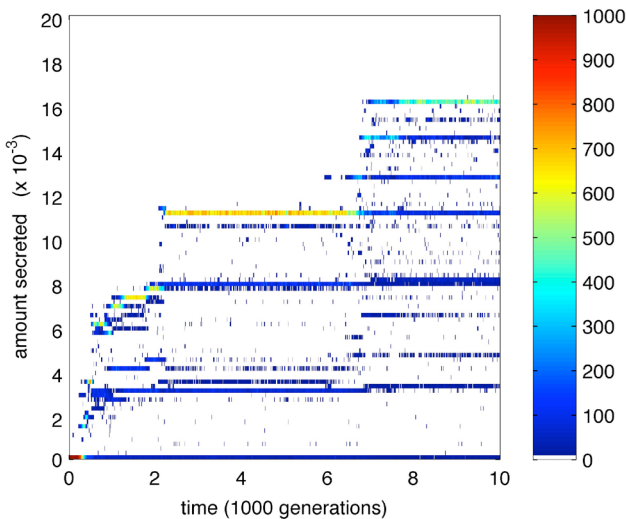


Figure 3. Frequency of different amounts of secretion in a single population over time. We binned organisms by the amount secreted into 100 equally sized bins between 0 and maximal secretion throughout the experiment and represented secretion by color. For clarity, any bin with less than 10 organisms is shown in white.

Migration, diffusion and degradation

Overall effects and interactions. As described in the methods, we performed experiments at several different levels of migration, diffusion and degradation. The evolutionary

	Sum sq.	d.f.	F	p
<i>mig</i>	0.0007	1	48.48	$< 10^{-4}$
<i>dif</i>	0.0009	1	65.28	$< 10^{-4}$
<i>deg</i>	0.0002	1	15.97	0.0001
<i>mig x dif</i>	0.0004	1	26.45	$< 10^{-4}$
<i>mig x deg</i>	0.0001	1	12.19	0.001
<i>dif x deg</i>	$< 10^{-4}$	1	0.09	0.7681
Error	0.0087	633		

Table 1. Three-way ANOVA of final average amount of public good secreted by an organism, with migration (*mig*), degradation (*deg*) and diffusion (*dif*) as independent variables.

trajectories of the amount of public good secretion are shown in Figure 4. In order to avoid statistical problems of multiple testing, we did not conduct a large number of t-tests to compare different treatments. Such tests would also be inappropriate for comparing trajectories of secretion over time, since these are repeated measurements on the same populations. Instead, to analyze the effect of different public good properties, we conducted a 3-way ANOVA on the average amount of public good secreted by an organism at the end of our experiments, with migration, degradation and diffusion as independent variables (Table 1). We find highly

significant effect of all three factors ($p \leq 0.0001$), indicating that migration, diffusion and degradation all strongly affect the level of secretion and thus cooperation that evolves in our experiment. Additionally, there are statistically significant interactions of migration with both other factors ($p = 0.0005$ for degradation and $p < 10^{-4}$ for diffusion) indicating the existence of complex interaction between these different features of cooperation. We did not find significant interactions between diffusion and degradation, which reflects the generally similar pattern of secretion within a column repeated across different columns in Figure 4, albeit with some scaling. We continue the analysis by examining the effects of each of the cooperation properties in greater detail.

Effect of migration rate. In this set of experiments we used $C = 0.03$, the cost of public good secretion that results in medium levels of cooperation in previous experiments. We found that an increased migration rate generally leads to lower levels of secretion (Figure 4), in accordance with existing theories on evolution of cooperation (Hamilton 1964; Crespi 2001; Kümmerli et al. 2009a). Effectively, cooperation is maintained because the organisms are more likely to be surrounded by kin when there is little or no migration.

However, there are scenarios under which the migration rate had no effect on secretion. For example, when diffusion is

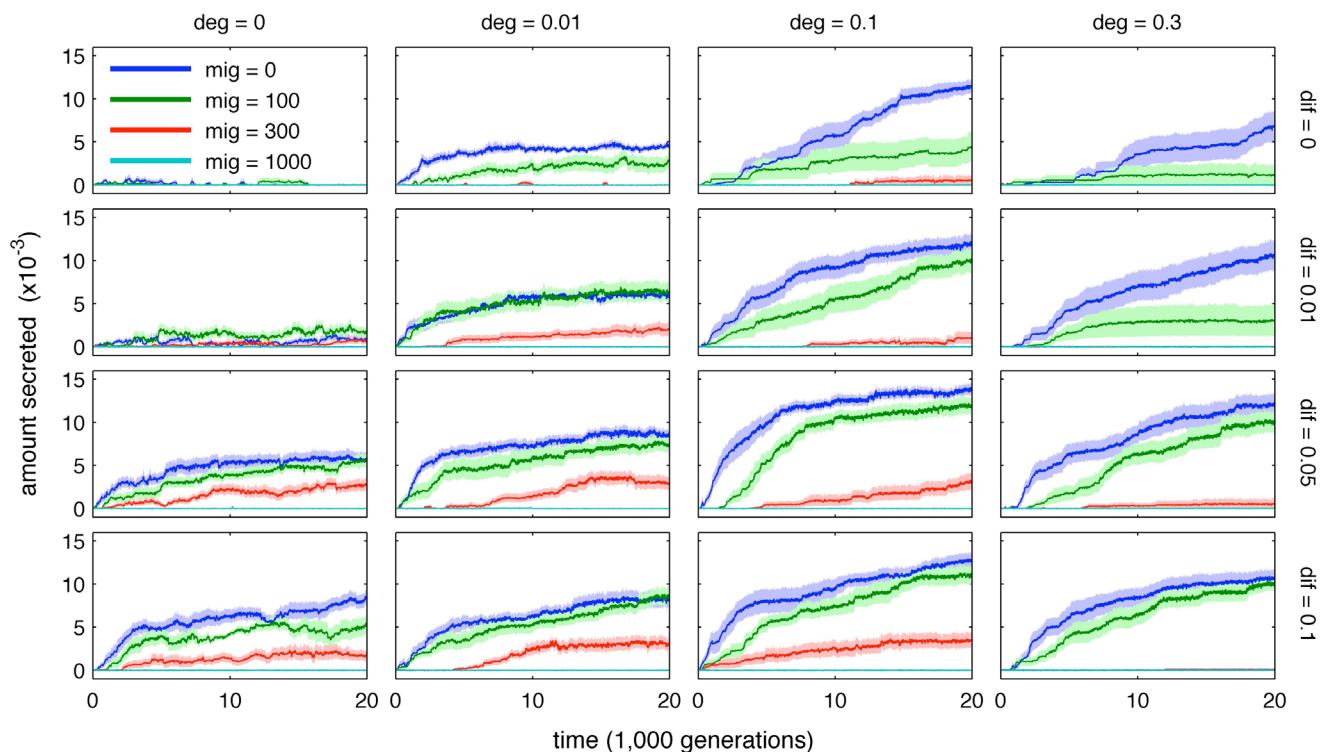


Figure 4. Effects of migration, diffusion and degradation on the amount of secreted public good over time. Each column of panels in this figure shows data from experiments with the same degradation rate (0%, 1%, 10% or 30% of the public good present degrades at the end of each generation). Each row of panels shows data from experiments with the same diffusion rate (0%, 1%, 5%, or 10% of the public good present in a grid cell diffuses into each of the 8 neighboring cells). Each line is an average of 10 experiments using the same set of parameters, with different colors representing different migration rates (in each generation 0, 100, 300, or 1000 pairs of organisms were chosen at random and their positions were swapped). The lightly shaded area around the lines represents one standard error of the mean.

0.01 and degradation 0.01, we observe identical secretion levels in experiments with migration of 0 or 100 (Figure 4). It is not the case that individuals are just secreting at the optimal rate for this set of conditions as the rate is continuously increasing over time. Another possible intuitive explanation is that at these lower levels of degradation secreted public good remains for a longer time, lowering the potential negative effect of migrating away from the public good one just secreted. Further detailed studies of evolutionary paths under specific conditions would be necessary to fully explain this aspect of our results.

Effect and interplay of diffusion and degradation. Our most unexpected and interesting result is the increase in the amount of secreted public good with the increase in degradation rate, across many levels of diffusion and migration (Figure 4). One direct and simple explanation is that organisms are effectively compensating the decrease in the amount of public good at their location by increasing their secretion. In other words, organisms secrete more in conditions where more public good is needed to achieve the same level of benefit via cooperation. We can test this hypothesis by analyzing the data on the average amount of public good that is present in a grid cell over time (Figure 5).

We find that indeed, all things being equal, there is less public good at higher rates of degradation, and that the highest secretion was unable to completely compensate and maintain equivalent contribution of cooperation to fitness. Although ultimately not fully successful in maintaining an unchanged amount of public good in the environment, such compensation

is still significant and constitutes a novel evolutionary strategy.

A somewhat different pattern emerges when instead of degradation we examine the effect of diffusion on the evolution of cooperation, both in terms of the average amount of the public good that is secreted per generation and of the average amount of public good that is present in the environment at a given time. Secretion increases as the percentage of public good that diffuses to neighboring cells changes from 0 to 1% and 5%, but it seems to decrease (or at best remains unchanged) when we compare the secretion curves for 5% and 10% secretion in Figure 4. Diffusion has a somewhat similar effect to that of degradation in so far as it decreases the amount of public good available locally. However, in case of diffusion, not only do the organisms compensate for the decrease by secreting more, but there is actually a greater amount of secreted compound present. We interpret the initial rise in diffusion as being beneficial to the evolution of cooperation because it modifies the surrounding locations in the population to be more suitable for being inhabited by cooperators – the public good diffusing there will be able to offset at least some of the secretion cost born by the organisms that inhabit these locations in the future. When smaller percentage of the public good degrades, a secreting organisms living in that exact location may compensate for the cost of secretion and survive to reproduce for another generation. In contrast, when public good diffuses to neighboring cells, it is enabling the spread of cooperators into those cells as well. Of course, at high diffusion rates (e.g.

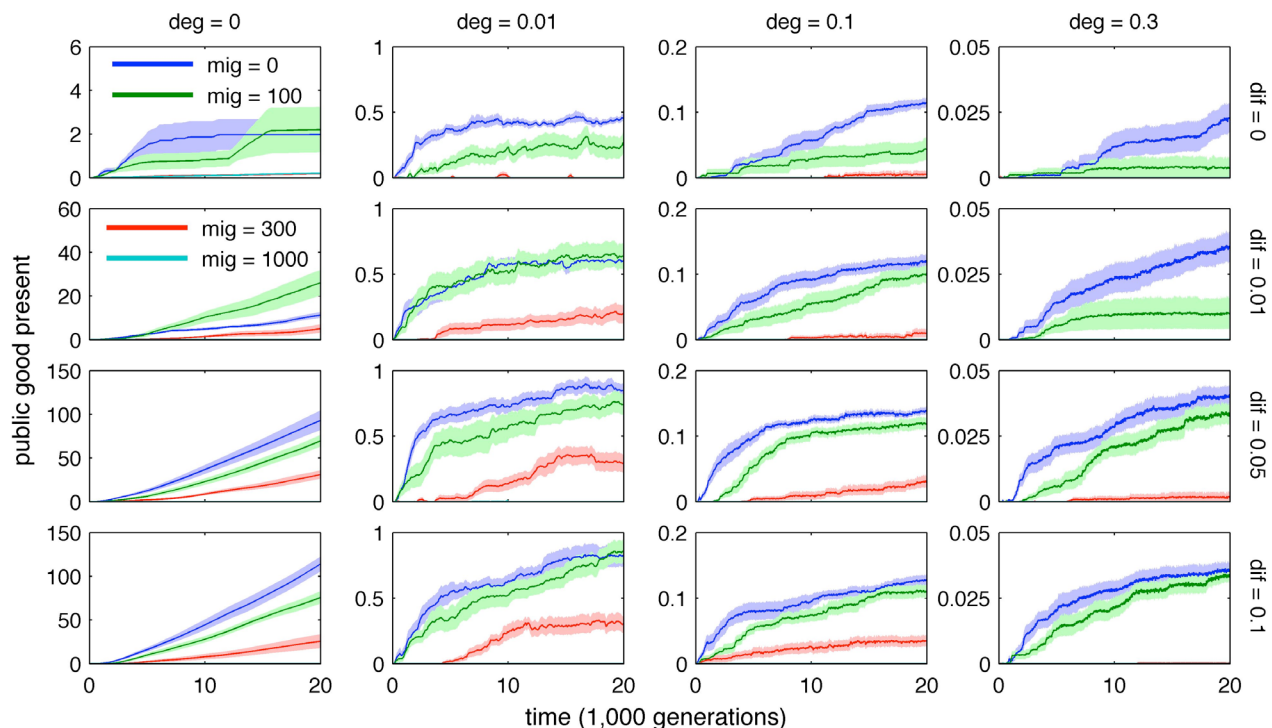


Figure 5. Effects of migration, diffusion and degradation on the amount of public good present in a grid cell over time. As in Figure 4, different colors represent different mutation rates, while columns and rows show data with the same degradation or diffusion rate, respectively. The light shading area around the curves represents one standard error of the mean. Note the different scales on the y-axes in different panels. For clarity, the legend for the entire figure is split between two panels.

5%), the public good may spread faster than the secretor genotype, benefiting unrelated and potentially non-secreting individuals, which in turn lowers the selection for cooperation.

Cooperation without diffusion or degradation. We included two limit cases in our experiments: scenarios where there is either no diffusion or no degradation of public good. While unlikely to occur in nature, these scenarios can be informative about relevant situations beyond the microbial world (e.g. organisms cooperating in producing fixed, large structures, such as termite nests or beaver dams). We find that organisms secrete less or not at all when degradation equals zero, even though their offspring remaining in the same location would receive a greater benefit from its parent's secretion, strengthening the (still indirect) selection for cooperation (Figure 4). Especially interesting is the case where there is neither diffusion nor degradation and little secretion goes a long way: while the amount of secreted public good in the population is higher than in many other cases (Figure 5, note the different scale for the y-axes), by the end of the experiment, there is no significant number of secreting organisms in any of the populations.

Degradation v. consumption of public good. In *Aevol* there is no explicit consumption of public good or any cost associated to it. However, given that the organisms occupy every single location in the population grid, we could consider the degradation rate to encompass both the actual decomposition of the molecules and their consumption by the individuals. This interpretation implicitly assumes that the rate of consumption is dependent on the amount of public good molecule present. While it would be interesting to examine the effect of these additional properties of public good systems, due to additional computational resources necessary, they remain outside of the scope of this study. We do not expect that the results would be qualitatively different, but separately considering different decreases in the amount of public good may uncover additional complexities of the evolution of cooperation.

Conclusions

In our study we introduced several new features of the *in silico* experimental system *Aevol* that enable the detailed study of the evolution and maintenance of cooperation. We implemented cooperation in spatially structured populations of digital organisms via a public good molecule that can be secreted (at a cost) and that degrades and diffuses over time. In our experiments we tested the effects of different levels of diffusion and degradation of the public good, cost of secretion, and organism migration on the evolved level of cooperation. We found a complex pattern of interactions between the degree of spatial organization and the physical properties of the cooperation mechanism. Most interestingly, we observed an increase in the secretion of public good at higher rates of diffusion and degradation. In nature many different cooperation systems can be observed, with much research being focused on microbial ones, based on producing and sharing a public good. There are equally many, if not

more ways of modeling cooperation dynamics *in silico*. The main message of our work is that we must carefully consider a greater range of physical properties that characterize both our digital and *in vitro/vivo* model systems to truly test hypotheses about the evolution of cooperation. Generic models of public good cooperation run the risk of remaining in a specific corner of parameter space, difficult to generalize and apply to natural systems. In our ongoing research we expand and improve the implementation of cooperation in *Aevol*, by introducing additional features such as multiple public good molecules, evolvable mechanisms for public good consumption, and mobile genetic elements, similar to bacterial plasmids. Each improvement in *Aevol* and related models continues to bridge the gap between complexity we observe in nature and the complexity we can capture with our computers, allowing us to make the continual advancements in understanding cooperative processes.

Acknowledgments. We wish to thank DevoLab and Lenski Lab at Michigan State University as well as the French National Institute for Nuclear and Particle Physics, Lyon, who provided the computational resources used for the experiments included here. We would also like to thank Guillaume Beslon, Carole Knibbe, the members of the Taddei team and INSERM Unit 1001 for many useful discussions, comments, and advice. We appreciated the careful reading and well-thought-out comments of the three anonymous reviewers who helped us improve the manuscript and provided a number of interesting directions for future research. Our work was financially supported by ANR COOP-INFO grant and EURYI Award to FT.

Contributions. DM conceived and designed the study, performed the experiments and wrote the paper. AF and DP contributed to necessary modifications of the *Aevol* code and helped with interpreting the findings. FT provided input on the study design, result analysis and interpretation.

References

- Ackermann M, Stecher B, Freed N, Songhet P, Hardt W-D, Doebeli M. 2008. Self-destructive cooperation mediated by phenotypic noise *Nature* 454
- Beslon G, Parsons DP, Sanchez-Dehesa Y, Pena JN, Knibbe C. 2010. Scaling laws in bacterial genomes: a side-effect of selection of mutational robustness. *Biosystems* 102: 32-40
- Brockhurst M, Buckling A, Racey D, Gardner A. 2008. Resource supply and the evolution of public-goods cooperation in bacteria *BMC Biology* 6: 20
- Brockhurst MA, Hochberg ME, Bell T, Buckling A. 2006. Character displacement promotes cooperation in bacterial biofilms. *Current Biology* 16: 2030-4
- Brown SP, Taddei F. 2007. The durability of public goods changes the dynamics and nature of social dilemmas. *PLoS One* 2: e593
- Crespi BJ. 2001. The evolution of social behavior in microorganisms. *Trends Ecol Evol* 16: 178-83

- Czaran T, Hoekstra RF. 2009. Microbial communication, cooperation and cheating: quorum sensing drives the evolution of cooperation in bacteria PLoS One 4: e6655
- Diggle SP, Griffin AS, Campbell GS, West SA. 2007. Cooperation and conflict in quorum-sensing bacterial populations Nature 411:414
- Dunny GM, Brickman TJ, Dworkin M. 2008. Multicellular behavior in bacteria: communication, cooperation, competition and cheating. BioEssays 30: 296-8
- Ellis RJ, Lilley A, Lacey SJ, Myrrell D, Godfray HCJ. 2007. Frequency-dependent advantages of plasmid carriage by *Pseudomonas* in homogeneous and spatially structured environments. The ISME Journal 1: 92-5
- Fiegna F, Velicer GJ. 2005. Exploitative and hierarchical antagonism in a cooperative bacterium PLoS Biol 3: e370
- Fletcher JA, Doebeli M. 2009. A simple and general explanation for the evolution of altruism. Proc R Soc B 276: 13-9
- Griffin AS, West SA, Buckling A. 2004. Cooperation and competition in pathogenic bacteria. Nature 430: 1024-7
- Hamilton WD. 1964. The genetical evolution of social behaviour. Journal of Theoretical Biology 7: 1-52
- Knibbe C, Fayard J-M, Beslon G. 2008. The topology of the protein network influences the dynamics of gene order: from systems biology to a systemic understanding of evolution. Artificial Life 14: 149-56
- Knibbe C, Mazet O, Chaudier F, Fayard J-M, Beslon G. 2006. Evolutionary coupling between the deleteriousness of gene mutations and the amount of non-coding sequences. Journal of Theoretical Biology 244: 621-30
- Kümmerli R, Brown SP. 2010. Molecular and regulatory properties of a public good shape the evolution of cooperation. Proc Natl Acad Sci USA 107: 18921-6
- Kümmerli R, Gardner A, West SA, Griffin AS. 2008. Limited dispersal, budding dispersal, and cooperation: an experimental study. Evolution 63: 939-49
- Kümmerli R, Griffin AS, West SA, Buckling A, Harrison F. 2009a. Viscous medium promotes cooperation in the pathogenic bacterium *Pseudomonas aeruginosa*. Proc R Soc B 276: 3531-8
- Kümmerli R, Jirny N, Clarke LS, West SA, Griffin AS. 2009b. Phenotypic plasticity of a cooperative behaviour in bacteria. J Evol Biol 22
- Le Gac M, Doebeli M. 2009. Environmental viscosity does not affect the evolution of cooperation during experimental evolution of colicogenic bacteria. Evolution 64: 522-33
- Lenski RE, Ofria C, Collier TC, Adami C. 1999. Genome complexity, robustness and genetic interactions in digital organisms. Nature 400: 661-4
- MacLean RC, Gudelj I. 2006. Resource competition and social conflict in experimental populations of yeast. Nature 441: 498-51
- Misevic D, Ofria C, Lenski RE. 2004. Sexual reproduction and Muller's ratchet in digital organisms. In Proceedings of Artificial Life IX, ed. JB Pollack, M Bedau, P Husbands, T Ikegami, RA Watson, pp. 340-5: MIT Press, Cambridge, Massachusetts
- Ofria C, Wilke CO. 2004. Avida: a software platform for research in computational evolutionary biology. Artificial Life 10: 191-229
- Parsek MR, Greenberg EP. 2005. Sociomicrobiology: the connections between quorum sensing and biofilms. Trends in Microbiology 13: 27-33
- Parsons DP, Knibbe C, Beslon GTC. 2010. Importance of the rearrangement rates on the organization of transcription. In Proceedings of ALife XII, ed. H Fellermann, M Dörr, MM Hanczyc, L Ladegaard Laursen, S Maurer, et al, pp. 479-86: MIT Press, Cambridge, MA
- Paton JC. 1996. The contribution of pneumolysin to the pathogenicity of *Streptococcus pneumoniae*. Trends in Microbiology 4: 103-6
- Queller DC, Ponte E, Bozzaro S, Strassmann JE. 2003. Single-gene greenbeard effects in the social amoeba *Dictyostelium discoideum*. Science 299: 105-6
- Ross-Gillespie A, Gardner A, Buckling A, West SA, Griffin AS. 2009. Density dependence and cooperation: theory and a test with bacteria. Evolution 63: 2315-25
- Strassmann JE, Zhu Y, Queller DC. 2000. Altruism and social cheating in the social amoeba, *Dictyostelium discoideum*. Nature 408: 965-7
- Velicer GJ, Yu YTN. 2003. Evolution of novel cooperative swarming in the bacterium *Myxococcus xanthus*. Nature 425: 75-8
- West SA, Griffin AS, Gardner A, Diggle SP. 2006. Social evolutionary theory for microorganism. Nature Reviews Microbiology 4: 597-607

Coevolutionary Dynamics between Roles and Social Sensitivity in an Extended Minority Game

Keita Nishimoto¹, Ivan Tanev², Katsunori Shimohara³, Reiji Suzuki⁴ and Takaya Arita⁵

^{1,4,5}Graduate School of Information Science, Nagoya University Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan

^{2,3}Faculty of Science and Engineering, Doshisha University, 1-3 Miyakodani, Tatara, Kyotanabe 610-0321, Japan

¹nishimoto@alife.cs.is.nagoya-u.ac.jp, ²itanev@mail.doshisha.ac.jp, ³kshimoha@mail.doshisha.ac.jp

⁴reiji@nagoya-u.jp, ⁵arita@nagoya-u.jp

Abstract

The social brain hypothesis suggests that humans evolved larger brains and intelligence as adaptations to an increasingly complex social environment. We believe that social role division is a key factor in the evolution of social intelligence. To examine the role of this factor, we extend Challet and Zhang's Minority Game by adding a pre-decision communication stage and using a continuous strategy space instead of a binary one, and develop an evolutionary model based on this game. The evolutionary simulations demonstrate that the system alternates between two states: one with homogeneous social behavior and the other with heterogeneous behavior. We observe differentiation of social roles in the latter state: we find a "pivotal agent" that tends to adopt low-risk, low payoff strategies but determines which strategy will be in the minority and which in the majority, and we find "risk taker" that tend to adopt high-risk, high pay-off strategies. Using social sensitivity as a measure of social intelligence, we show that the level of social sensitivity correlates with the social roles, and is also a major factor in the mechanisms by which social roles switch.

Introduction

Primates, including humans, have relatively large brains and more highly developed intelligence than other mammals. However, the question why and how they acquired this high intelligence remains unsolved. In recent years, the "social brain hypothesis" has attracted the attention of researchers trying to explain the evolution of human intelligence. The hypothesis claims that primates' large brains reflect the computational demands of their complex social environment (Dunbar, 1998), and that social conflicts played an important role in the evolution of primate intelligence (Chance and Mead, 1953).

When we consider the social interactions in especially humans, we see that "role" is key ingredient. The richness and importance of roles in human society is outstanding compared to the other primates (Wilson, 1975). Moreover, humans can switch roles dynamically in response to varying social situations. Our highly developed social intelligence is speculated to be necessary for us to act appropriately in response to observations of others in a large variety of social situations.

The purpose of this study is to clarify the relationship between role differentiation/switching and social intelligence from a coevolutionary perspective. Specifically, we perform evolutionary simulations using the Dynamic Minority Game (DMG) to investigate the mechanisms of the emergence and dynamic switching of social roles, and the relationship between social intelligence and role division.

The Minority Game (MG), initially proposed by Challet and Zhang (Challet and Zhang, 1997), is a minimalist econophysics platform that captures a common social scenario. In each round, N (odd) agents independently choose between two options, and those who have selected the least selected option (i.e. the minority side) win and are awarded a point.

DMG is an extension of MG in two aspects. 1) Agents select a strategy from a continuous space instead of two alternatives, and 2) a pre-decision communication stage is incorporated. These modifications make it possible for DMG to express 1) a kind of social role division, and 2) a dynamic decision making process involving negotiation between agents, respectively.

Our evolutionary simulations show that the system alternates between three phases. We see a differentiation of social roles in one of the phases: we find a "pivotal agent" that tends to adopt a low-risk, low payoff strategies but determines which strategy will be in the minority and which in the majority, and we find "risk takers" that tend to adopt a high-risk, high pay-off strategies. This paper focuses on the role of social sensitivity (a measure of social intelligence) dynamics in the transitions between phases and in role switching mechanisms.

Dynamic Minority Game (DMG)

In the Minority Game with N agents proposed by Challet and Zhang, the *payoff* of an agent i choosing alternative A_i is calculated as follows:

$$\text{payoff} = -A_i \text{sgn}\left(\sum_{k=1}^N A_k\right), \quad (1)$$
$$(A_i \in \{-1, 1\}, \quad \text{payoff} \in \{-1, 1\}),$$

$$\text{sgn}(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x = 0, \\ -1 & \text{otherwise.} \end{cases}$$

We propose a Dynamic Minority Game by extending the Minority Game on the following two points. First, we adopt a continuous strategy space instead of a binary one. The payoff of agent i with strategy value a_i is calculated as follows:

$$\begin{aligned} \text{payoff} &= -a_i \text{sgn}\left(\sum_{k=1}^N \text{sgn}(a_k)\right), \\ (a_i \in [-1, 1], \quad \text{payoff} &\in [-1, 1]), \\ a_i, \text{payoff} &: \text{real value.} \end{aligned} \quad (2)$$

This equation represents the situation as follows: The possible signs of the strategy value (positive or negative) correspond to the alternatives in the standard Minority Game. Agents win the game if the sign of their strategy is the minority sign in the group. Furthermore, the strategy's absolute value defines its "intensity". Higher intensity values lead to both higher risk and higher reward. The winning agents obtain a positive payoff equal to the absolute value of their strategies.

Secondly, we add a pre-decision communication stage before the agents confirm their strategy. During this stage, agents can continuously adjust their strategy. The tentative strategy of agent i at time step t ($= 0, 1, \dots, T-1$) is represented as $a_i(t)$ ($a_i(0) = 0$). Each agent can adjust $a_i(t)$ gradually by $\epsilon(t)$, after observing others' tentative strategies in the previous step. The final decision of agent i : a_i is defined as $a_i(T)$, and used for calculation of payoffs (3).

$$\begin{aligned} a_i &= a_i(T), \\ a_i(t+1) &= a_i(t) + \epsilon(t), \\ (t &= 0, 1, \dots, T-1). \end{aligned} \quad (3)$$

Note that if $a_i(t) + \epsilon(t) > 1 (< -1)$, then $a_i(t+1) = 1 (-1)$. In this study, we focus on the case of $N = 3$, the minimum number of agents for a Minority Game. Figure 1 shows an example game. The x-coordinate corresponds to the time step t and the y-coordinate represents $a_i(t)$ for each agent.

Model

Mechanism of decision making

Every agent is equipped with a Recurrent Neural Network (RNN) to decide $\epsilon(t)$ at each step. The reason why we choose to use Recurrent NNs is to enable agents to make decisions appropriately depending not only on the current inputs, but also on past inputs: RNNs can use their internal memory to process arbitrary sequences of inputs. Each

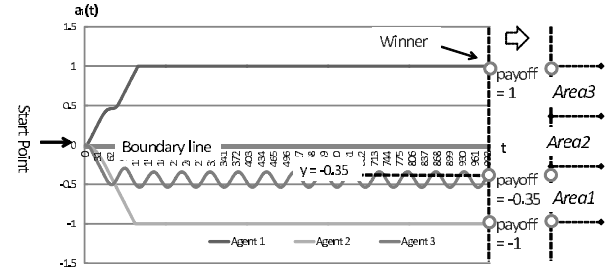


Figure 1: A trial of DMG ($N = 3, T = 1000$).

RNN has three layers (5 input units, 6 hidden units, 4 output units), and the units use a sigmoid activation function ($f(x) = 1/(1 + \exp(-x))$). For simplification of the model, RNNs do not have bias units. Two output units in the output layer are recurrently connected to two input units in the input layer.

Every time step, the agent's RNN receives five input values: its own current strategy value, the distance from the strategy values of the other two agents to its own, and the values from the two output units from the previous step. Two units in the output layer generate the values au and ad , which determine $\epsilon(t+1) = \epsilon(t) + (au - ad)/100$. The remaining two output units are connected one-to-one to two of input units.

Evolutionary Algorithm

The full set of 54 connection weights in each RNN is encoded in the genotype of each agent and evolved using a simple type of Evolutionary Strategy (ES). The connection weights do not change during a trial. We assume three independent gene pools, each of which provides one agent in each trial, so the agents that interact in a game trial come from independently evolved gene pools. Each gene pool has N_p individuals. At the beginning of each generation, we randomly assemble N_p groups of three individuals, one from each pool. Then, one trial of Dynamic Minority Game is played in each group. This procedure of group assembly and game trial is repeated R times. The fitness of each individual is defined as accumulated payoff over R trials. The population in the next generation of each gene pool is composed as follows: First, we select n best individuals from N_p individuals (the elite), and preserve them to the next generation. Then, each of elite contributes two copies of themselves to the next generation, and small random values from a normal distribution with a fixed standard deviation are added to each connection weight in the offspring. Finally, $N_p - 3n$ individuals with randomly generated genotypes are added to the population. These evolutionary operations for selection and reproduction are performed on each gene pool independently.

Evaluation of social sensitivity

In order to be able to track evolution of social intelligence in this model, we define a measure of “social sensitivity” of an agent, which estimates the degree to which the focal agent responds sensibly to the others’ strategy values. In a DMG with three agents, the optimal strategy value of an agent that maximizes its expected payoff can be derived from the strategy values of the other two agents under the assumption that all agents adopt their current tentative strategy at the last time step. This optimal strategy will be -1, 0, or +1. When the strategies of the other two agents are positive (negative), the optimal strategy of the focal agent is -1 (+1). When the signs of the other agents are different, the optimal strategy is 0. If an agent tends to keep its strategy close to the optimal value, we interpret this as an indication that the agent makes a decision based on the observation of others, in other words, the individual has high social sensitivity.

Specifically, we divided the strategy space at each step into three areas: area 1 = [-1, -0.33], area 2 = [-0.33, +0.33] and area 3 = [+0.33, 1], corresponding to the optimal strategy values of -1, 0, and 1 respectively. We prepared some pairs of test agents with fixed predefined behavioral patterns, to serve as static social environments for measuring the social sensitivity of an agent. The social sensitivity is defined as the average proportion of steps during which the agent’s strategy value is in the optimal area. A preliminary analysis showed that agents that ignore others’ behavior score a social sensitivity value of about 0.3, agents that consider one of the other agents’ behavior score about 0.5, and agents that observe and respond appropriately to the behavior of both of the other two agents reach a sensitivity score between 0.8 and 0.9.

Results

Evolution of agent’s behavior

We evolved the population for 10000 generations. We used the following parameter settings: $T = 1000$, $N_p = 40$, $n = 12$, $R = 40$. Initial connection weights are drawn randomly from a uniform distribution over $[-1, 1]$, and mutation adds a random number from the normal distribution $N(0, 0.2^2)$. Although we confirmed that the results did not vary qualitatively even if using different parameters, it was observed that the whole system becomes more stable when we set the standard deviation of the normal distribution N lower. The details will be described later.

First, we focus on how behavior and social sensitivity develop during the early stages of the evolution process. Figure 2 represents the average fitness of each gene pool and the average fitness of all individuals from the 0th generation to the 99th generation. We see a rapid increase of fitness in all gene pools. The average fitness reached to approximately -5 at the 99th generation. Figure 3 shows the evolution of social sensitivity for the same experiment depicted in Figure 2. We

see that social sensitivity increased gradually, but there are significant differences between gene pools. We will return to this point later.

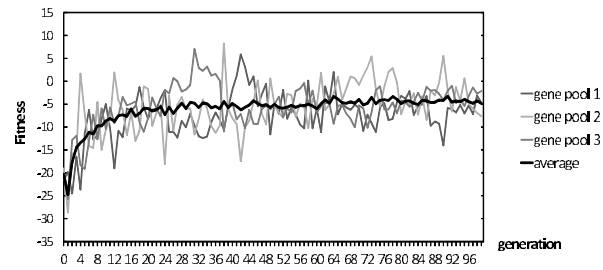


Figure 2: Evolution of fitness.

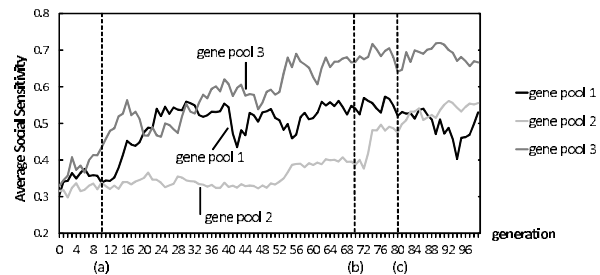


Figure 3: Evolution of social sensitivity.

Figure 4 shows an example behavior at the 10th generation, when the social sensitivity was still low. We see that the strategy value of one agent reached the upper limit and that of another agent reached the lower limit, while the remaining agent’s value remained near the boundary line ($a_i(T) = 0$). We focus on the agent whose strategy value remained near the boundary line. In the situation shown in Figure 4, the strategies of the other two agents are on the upper and the lower area respectively, and they did not change their strategy values. Thus, the focal agent could not avoid ending up on the majority side, and so its payoff falls below 0 regardless of which side it picks. The optimal behavior thus is to choose a strategy value as close to the boundary line as it can, and receive payoff of near 0. It was often observed in the simulations that the final strategies of the three agents settled on these three positions on the strategy space: the upper limit, around the boundary line, and the lower limit. Such differentiation of behavior as observed in Figure 4 is expected to appear often because diversity of strategies is essential to good performance in minority games.

The average payoff of each agent over R trials becomes near 0, if 1) the final strategy of the agent nearby the boundary line stays very close to it, and 2) its sign splits fifty-fifty between positive and negative. In this scenario, the situa-

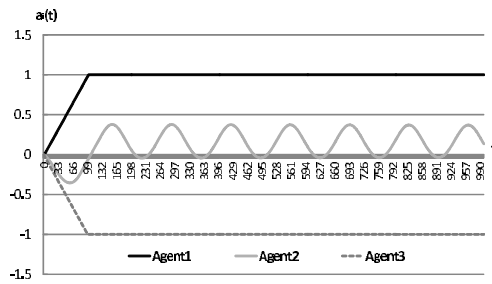


Figure 4: Behavior of agents at the 10th generation.

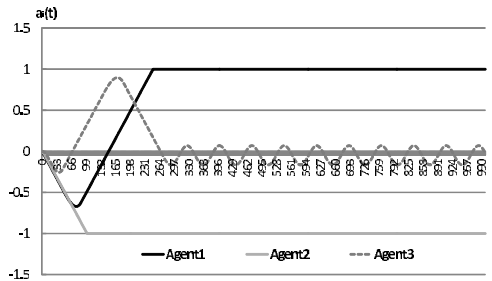


Figure 5: Behavior of agents at the 70th generation.

tion where the final strategies fall in the same area of the strategy space (yielding payoff far below 0) is avoided. This scenario is speculated to be a kind of equilibrium for each agent. Each of these three areas on the strategy space can be said to correspond to a “role” in avoiding situations that are disadvantageous to all. Thus, the observed distribution of strategy choice can be interpreted as a form of “role differentiation” between agents. In Figure 2 we see an increase in average fitness from the initial generation to the 20th generation, likely due to the emergence of this role differentiation. We refer to the two agents who choose strategies at the upper and lower limit as “risk taker” because they aim for high profit at high risk. We refer to the agent who stays near the boundary as a “pivotal agent” because the winner is decided by the sign of its strategy value. In addition, “pivotal agent” is considered to play a crucial role in maintaining the role division structure. If pivotal agent’s strategy is biased to one side, the fitness of risk taker on that side decreases. As a result, the risk taker cannot but change its risk-taking strategy, breaking the stable role division.

By the 70th generation, gene pool 1 (for Agent 1) and of gene pool 3 (for Agent 3) had both evolved high social sensitivity as shown in Figure 3 (b). Figure 5 shows a representative game from the 70th generation in which Agent 1 and Agent 3 can be seen to respond each other’s behavior. In Figure 5, all agents initially lower their strategy values. Agent 3 (at around step 40), and Agent 1 (at around step 80) can be seen to switch direction and start increasing

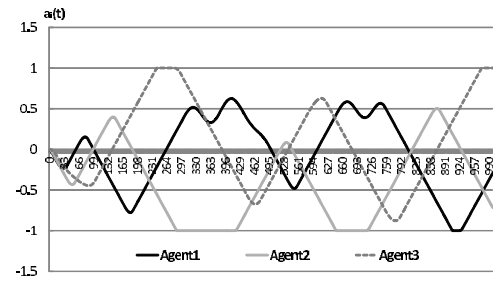


Figure 6: Behavior of agents at the 80th generation.

their strategy values, in order to avoid the situation where the strategy values of all agents remain negative and all lose. Once the strategy value of Agent 1 surpasses 0 (at around step 150), Agent 3 switches direction again in response. The most likely explanation for these behaviors of Agent 1 and Agent 3 is that they changed the increase or decrease in their strategy values in response to the strategy values of others.

Figure 6 shows the behavior of agents at the 80th generation. We can observe that agents interacted with each other more actively than was the case in the earlier generations. In Figure 3 (c), we see that all gene pools acquired relatively high social sensitivity at the 80th generation, which is likely the cause of the increased fluctuation of agents’ strategy values. This sort of fluctuation was often observed in the simulation.

Evolution of social sensitivity

Figure 8 shows example behavior of an individual agent that appeared during the evolutionary simulation. This agent has a social sensitivity score of 0.9. We show its behavior in one of our static test environments. We observe that the agent adjusts its own strategy appropriately in response to the positions and movements of the other agents’ strategy values.

Figure 7 (A) shows the average social sensitivity of individuals in each gene pool from the 1100th generation to the 6330th generation in the evolutionary simulation. It is noteworthy that not necessarily all of individuals reached high social sensitivity, and that social sensitivity varied per gene pool. We divide the transition of the social sensitivity shown in Figure 7 (A) into three phases as follows:

Phase 1: One gene pool evolves high social sensitivity, while the sensitivity of the other two pools remains lower (Figure 7 (1), (3), (6), (8), (10)).

Phase 2: Two pools evolve high social sensitivity, while the sensitivity of the remaining one remains low (Figure 7 (2), (4), (7), (9)).

Phase 3: All of the pools evolve high, approximately identical social sensitivity levels (Figure 7 (5)).

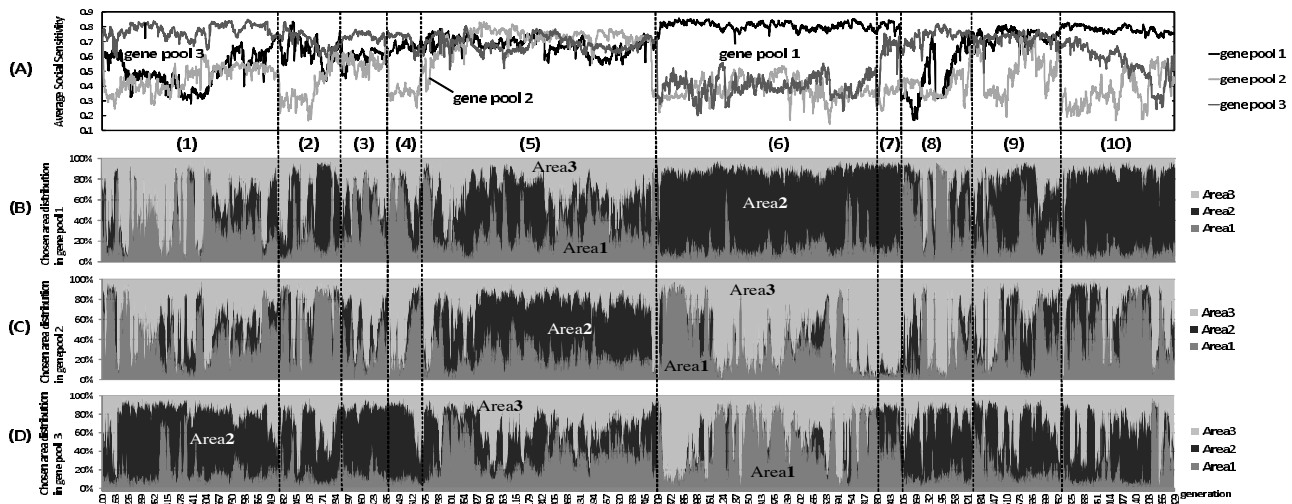


Figure 7: (A) Average social sensitivity of each gene pool in 1100th ~ 6330th generation. (B) ~ (D) Chosen area distribution in each gene pool in 1100th ~ 6330th generation.

These phases spontaneously switched through the generations. When we set the standard deviation of normal distribution N low, the period of Phase 1 tends to increase, and that of Phase 2 and 3 tends to decrease. Also, it was observed that the frequency of phase switch became low.

The differences in social sensitivity level between gene pools are closely related to the social roles (i.e. strategies) the pools adopt. We will return to this point below.

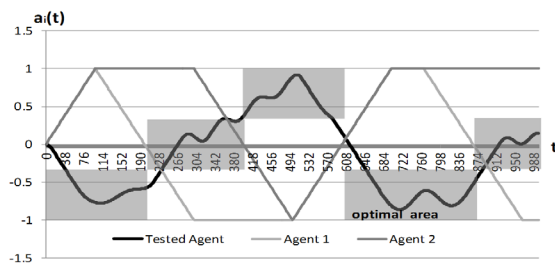


Figure 8: Behavior of the individual with social sensitivity 0.9 in a static environment (The areas filled with gray are the optimal areas.).

Role differentiation between gene pools

In this section, we examine whether there is behavioral differentiation between gene pools. We divide the final strategy space into three areas, and label them Area 1 ~ 3 from the bottom to top (see Figure 1). Area 1 and Area 3 correspond to the risk taker's strategies, and Area 2 corresponds to the pivotal agent's strategy. Figure 7 (B), (C) and (D) show each pool's strategy choice distribution over these ar-

reas, from the 1100th to the 6330th generation. These reveal the characteristics of the individual gene pools, showing that significant between-pool differentiation occurs in behavioral tendencies. For instance, Figure 7 (6) shows that from the 3840th generation to the 4850th generation, the individuals in gene pool 1 picked strategies in Area 2 with high probability and rarely strategies in Area 1 or Area 3. On the other hand, in gene pool 1 and gene pool 3 we see a strong tendency to pick strategies in Area 1 and Area 3 over strategies in Area 2. In other words, role differentiation (risk taker / pivotal agent) occurred between gene pools. As with social sensitivity, we can distinguish two phases with respect to role differentiation: one with clear role differentiation between pivotal agent and risk takers, and one without clear differentiation. We elaborate on this observation in the next section.

Relationship between roles and social sensitivity

Here we focus on the observation that the role differentiation phases and the social sensitivity phases often shift at the same time. Looking at Figure 7 we can see that the state of the gene pools alternated between Phase 1 (clear role differentiation and social sensitivity differentiation) and Phases 2, 3 (unclear role differentiation). Figure 9 shows the relationship between social sensitivity and strategy area. We see that the social sensitivity of individuals that tend to choose strategies in Area 2 is higher than that of the other individuals.

To clarify this relation, we focus on Phase 1. In this Phase, we see a clear differentiation both in roles and social sensitivity across gene pools, with high and low social sensitivity correlating strongly with the tendency to reach Area 2 or Ar-

as 1 or 3, respectively. In other words, there is a clear role division: a pivotal agent with high social sensitivity and two risk takers with low social sensitivity.

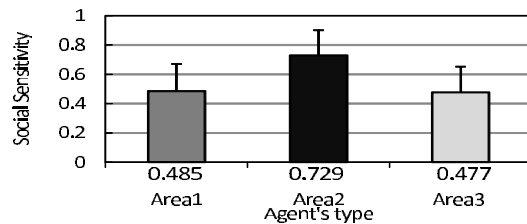


Figure 9: Agent's type and social sensitivity.

The reason for this would be as follows. Depending on the sign of the final strategy of the pivotal agent, the risk takers can attain an average payoff per game above 0. However, the average payoff of the pivotal agent is always slightly below 0, because the pivotal agent always ends up on the majority side. Therefore in order to maximize its payoff, the pivotal agent must pay closer attention to others' actions, which evolves social sensitivity of pivotal agent.

The analyses in this section make it clear that there is a high correlation between the social sensitivity and the social roles of the gene pools.

Role switch between gene pools

So far, we have divided the roles into pivotal agent and risk takers. However, role differentiation also occurs between gene pools which play the role of risk takers. We now focus on role switching between risk takers (Figure 10 (A)).

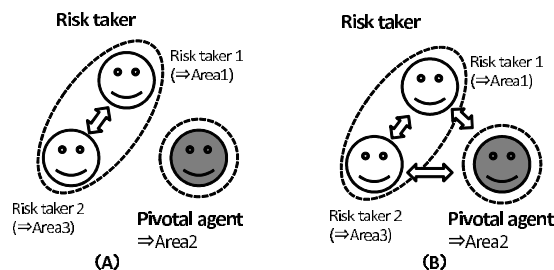


Figure 10: (A) Role switch between risk takers pools. (B) Role switch between three gene pools.

Figure 11 shows the social sensitivity of each gene pool, and the distribution of strategy choice over the three areas for gene pool 1 and gene pool 3 from the 8818th generation to the 8838th generation. We see that at the 8818th generation, individuals from gene pool 1 mostly pick strategies in Area 1, and that individuals from gene pool 3 mostly pick strategies in Area 3. That is, individuals from both pools play the risk-taking role. The fact that the pools consistently

pick complementary risk taking roles can be regarded as role differentiation between these two pools. Figure 11 shows two switches in this role division, one over the course of generations 8820 ~ 8822 and one over the course of generations 8830 ~ 8834. Role switching between risk taker gene pools occurred often in Phase 1. It is noteworthy that Phase 1 remains stable over such a role switch.

The mechanism of role switching between risk takers gene pools can be summarized as follows:

1. In the gene pool with the lower social sensitivity (gene pool 3 in Figure 11), mutation causes the appearance of a mutant offspring with a different behavioral tendency from the rest of its pool. This new behavioral tendency spreads in the gene pool if by chance the mutant individual gets good payoffs.
2. The average social sensitivity of the other gene pool (gene pool 1 in Figure 11) is 0.5. This means that individuals in this pool are to some extent sensitive to others' behavior. Therefore, when a mutation causes a sudden change of behavioral tendency in the other pool, this pool can switch its behavior in response.

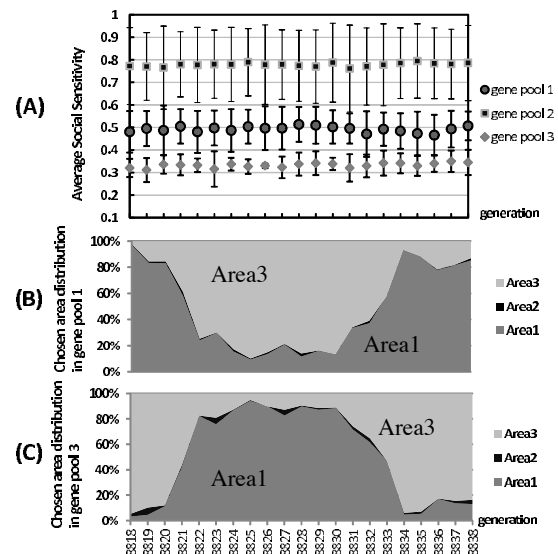


Figure 11: (A) Average social sensitivity of each gene pool. (B),(C) Chosen area distribution in the gene pools 1 and 3.

In short, this type of role switching results from mutation-caused change of the behavioral tendency of a gene pool with low social sensitivity, and the subsequent social sensitivity-based adaptation to that change by the gene pool with higher social sensitivity. In other words, social intelligence enables individuals to switch their roles flexibly and dynamically when others' behavior suddenly changes. If, on the other hand, behavioral tendency changes in the gene pool

with higher social sensitivity, the individuals in the other gene pool, having low social sensitivity, cannot adapt. Consequently in this situation, role switching is expected not to occur.

In Phase 1, social sensitivity of the three gene pools often settles around $0.8 \sim 0.9$, 0.5 , and 0.3 . This distribution of social sensitivity is thought to be a robust configuration that stabilizes the whole system, even if the behavioral tendency of the gene pool with the lowest social sensitivity suddenly changes due to mutation.

We conclude from this analysis that social intelligence plays an important role for role switching between gene pools. Next, we focus on the role switching between three gene pools (Figure 10 (B)). Figure 12 (A) shows the social sensitivity of each gene pool from the 3767th generation to the 3880th generation. Figure 12 (B) ~ (D) shows the distribution of strategy choice over the areas for each gene pool. We see that social sensitivity shifts phase from Phase 3 to Phase 2, and from Phase 2 to Phase 1. We also see a transition period from an ambiguous state to a stable state with clear role differentiation.

The transition process can be explained as follows:

1. Figure 12 shows that around the 3778th generation (i), gene pool 2 saw a rapid decrease in social sensitivity (A), and simultaneously a rapid increase in the proportion of individuals picking strategies in Area 3 (C), indicating a shift towards picking strategies in Area 3 without observing the behavior of the other agents. This constitutes a phase shift in the global social sensitivity configuration from Phase 3 to Phase 2. It was observed several times in our simulations that the sort of role switch as occurs at time (i) is caused by a decrease in social sensitivity.
2. In response to this change in gene pool 2, the individuals in gene pool 1, who tended to pick strategies in Area 3 until the 3778th generation, switch their choice of strategy areas to Area 1.
3. Then in response to this change in pool 1, individuals in gene pool 3 switch their strategy choice from Area 1 to Area 2. This restores global stability, which is then maintained for a while.

We can regard this process as a chain reaction of role switches across the three gene pools triggered by a behavioral mutation in one gene pool (pool 2 in this case). Next we look at the transition process starting around generation 3806 (Figure 12 (ii)).

4. A decrease in social sensitivity occurs in gene pool 3 over the 3806th to the 3809th generation, and the individuals in gene pool 3 come to pick strategies in Area 1. The decrease in sensitivity constitutes a shift in global social sensitivity configuration from Phase 2 to Phase 1. In response to the behavior change in gene pool 3, the individuals in gene pool 1 switch their strategy choice from Area

1 to Area 2. This completes a role switch between gene pool 1 and gene pool 3.

5. Subsequently, a role switch occurs between pool 2 and pool 3 (a role switch between risk taker pools) occurring gradually over a relatively long time span.

Finally, the population reaches a stable state where the strategy choice in gene pool 1 is in Area 2 (pivotal agent role), the strategy choice in gene pool 2 is in Area 1, and the strategy choice in gene pool 3 in Area 3 (risk taker role). Social sensitivity of the gene pools at the 3880th generation was 0.82 , 0.32 , and 0.46 , an instance of the robust configuration mentioned above.

Figure 12 (A) shows that past the 3806th generation the individuals in gene pool 1 came to have a tendency to primarily pick strategies from Area 2. At the same time, the social sensitivity of gene pool 1 starts to increase gradually after the 3806th generation. This means that social sensitivity in gene pool 1 evolved to high values as its individuals played the pivotal agent role.

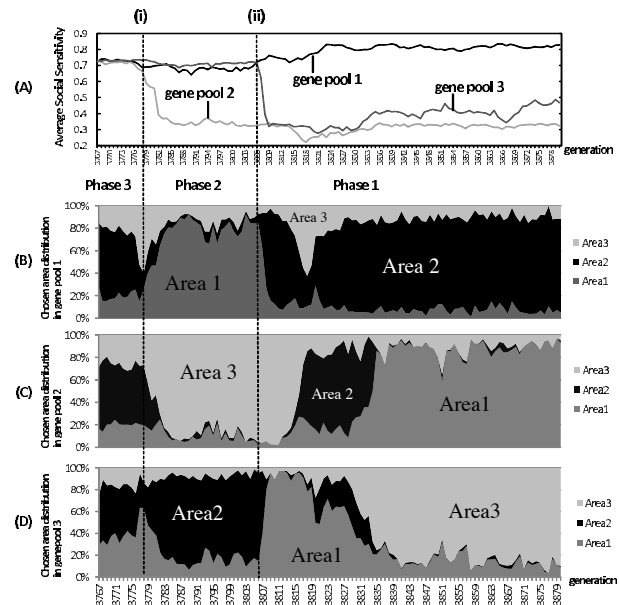


Figure 12: (A) Average social sensitivity of each gene pool. (B) ~ (D) Chosen area distribution in each gene pool.

Conclusions

In this paper we introduced the Dynamic Minority Game as an extension of the standard Minority Game, and used it to investigate the mechanisms of the emergence and dynamic switching of social roles, as well as the relationship between social intelligence and social role, in a computational model. We defined a social sensitivity measure as a means to track the social intelligence and evaluated the dynamics of the

model. We found that the system switches between three phases, characterized by the difference in global social sensitivity configuration over three gene pools each of which provides one agent for the three-player Dynamic Minority Game. When the difference in the social sensitivity between the gene pools is large, role division tends to be stable. In one of the phases two distinct roles emerge: we see a “pivotal agent” with high social sensitivity adopting primarily low-risk, low payoff strategies but with its choice determining the payoff outcomes of the other agents, and two “risk takers” with low social sensitivity adopting primarily high-risk, high payoff strategies.

We then focused on the mechanism of the transitions between system phases, and on transitions between social roles. It was shown that social sensitivity plays a critical role in both transitions. We observed that sudden mutation-induced behavior changes in one pool can be compensated by other, socially sensitive pools via a role switch.

We note three points of particular interest in our results. First, we observed that agent behavior evolves in tandem with social sensitivity and that our evolutionary model captures such aspects of social behavior as differentiation of social role and dynamic fluctuations therein. Agents observe each others’ strategy values, and change their own strategies accordingly, so we can regard the agent’s behavior in pre-decision communication stage as a type of *signal*. The evolution of animal signals has been studied by many researchers (Smith and Harper, 2004). Our observation that signals change with the evolution of social intelligence is expected to provide new perspectives to these researches.

Secondly, we saw that roles are differentiated most clearly when there is large variation in social sensitivity between pools. This result suggests that fixation of social role division over the pools causes large differences in social intelligence between pools. Moreover, it indicates that the difference in social sensitivity helps to fixate social roles and stabilize society. Andrew et. al suggest that character differences are crucial for the emergence of leadership and followership, and that, they are maintained in populations because they foster social coordination (King et al., 2009). If we view our social sensitivity measure as a character trait, then our results support strongly this hypothesis. From the view of leadership, the pivotal agent can be described as a leader in three agents as we suggested the probability that pivotal agent plays an important role to stabilize the role division structure. Research on leadership has shown individuals who are bold and do not care about others are likely to become leaders (Conradt et al., 2009)(Vugt, 2006). However, in our experiments, the individual with highest social sensitivity becomes a leader, which provides a different perspective.

Thirdly, we saw that role switching can be initiated via a decrease in social sensitivity, or a change of behavioral tendency in a pool with low social sensitivity, and then com-

pleted by the adaptive behavior of individuals with higher social sensitivity. Drea and Carter conducted experiments to investigate cooperative behavior in pairs of spotted hyenas (Drea and Carter, 2009). When a naive animal unfamiliar with the task was paired with a dominant experienced animal, it was observed that the dominant one switched its social role, and adjusted its behavior to the naive one to accomplish the task. This result is similar to our results in that the roles are switched by the adjustment of the more capable agents to the less capable ones. It is noteworthy that the role switching dynamics we observed in a competitive setup so resemble the role switching that Drea and Carter observed in cooperative behavior.

References

- Challet, D. and Zhang, Y.-C. (1997). Emergence of Cooperation and Organization in an Evolutionary Game. *Physica A*, 246-Issues 3-4:407,418.
- Chance, M. R. A. and Mead, A. P. (1953). Social behavior and primate evolution. *Symposia of the Society for Experimental Biology*, 7:395,439.
- Conradt, L., Krause, J., Couzin, I. D., and Roper, T. J. (2009). “Leading According to Need” in Self-Organizing Groups. *The American Naturalist*, 173:304,312.
- Drea, C. M. and Carter, A. N. (2009). Cooperative problem solving in a social carnivore. *Animal Behaviour*, 78:967,977.
- Dunbar, R. I. M. (1998). The Social Brain Hypothesis. *Evolutionary Anthropology*, 6:178,190.
- King, A. J., Johnson, D. D. P., and Vugt, M. V. (2009). The Origins and Evolution of Leadership. *Current Biology*, 19:911,916.
- Smith, J. M. and Harper, D. (2004). *Animal Signals*. Oxford University Press.
- Vugt, M. V. (2006). Evolutionary Origins of Leadership and Followership. *Personality and Social Psychology Review*, 10:354,371.
- Wilson, E. O. (1975). *Sociobiology: The New Synthesis*. Harvard University Press.

Characterizing Autonomy in the Web via Transfer Entropy Network

Mizuki Oka¹ and Takashi Ikegami²

¹Center for Knowledge Structuring, The University of Tokyo, Hongo, Tokyo 113-8656 Japan

²Graduate School of Arts and Sciences, The University of Tokyo, Komaba, Tokyo 153-8902 Japan

Abstract

Using the idea of transfer entropy (TE), we study autonomy and information flow on the Web and the newly defined TE network. The Web shows rich and complex autonomous network dynamics. Social network services (e.g., Twitter or Facebook) are now becoming a major source of Web dynamics in addition to the Web search services (e.g., Google). It is widely accepted that Twitter messages (called "tweets") and Google search queries react strongly to significant social movements and accidents, which are often characterized by bursting patterns in the time sequences. We call this the *reactive mode* of the Web. On the other hand, the Web dynamics, without the significant social events, seem to have an intrinsic rich dynamics, which we call the *default mode* of the Web. In this paper, we study the default mode of the Web system, which we characterize via a TE network. The amount of information flow transferred between different sequences of Google queries as well as Twitter keyword frequencies is investigated and we compute a TE network among Twitter keywords. We then discuss that the default mode of the Web can be characterized by the "breathing" dynamics of the TE network over a scale of a few weeks. We further use this idea of the default mode to install autonomy into generic artificial life systems.

INTRODUCTION

The difference between the study of artificial life and artificial intelligence is the way that autonomy is dealt with. We may be able to make an artificial intelligent system by using a large database with a very fast CPU, but such a system will not acquire autonomy in the same way that we find among living systems in general. What happens if autonomy comes first and we assume that intelligence merely emerges as a side effect of living systems (Ikegami, 2012)?

A simple but primary definition of an autonomous system is that it is a non-reaction system. A simple reactive system is characterized by action selection, which is given most likely as a function of the external stimuli to the system. An autonomous system must select its action by itself, but a system that is always indifferent to the external stimuli is, again, non-autonomous in the sense of living systems. It is just a decoupled random behavior from the environment.

Therefore, the Brownian particle (as an example of a reactive system) and chaos dynamics (as an example of a system indifferent to the external stimuli) are not biologically autonomous systems.

We thus propose that biological autonomy must be created between a system and its environment. A system must temporarily couple and decouple with the environment via the system's internal states; that is, a system sometimes, but not always, responds to the external stimuli. A concrete example of such autonomous dynamics is found in the embodied chaotic itinerancy (Ikegami, 2007) - a high-dimensional transition dynamic among pseudo-attractors that couples with the environment. Using chemical materials, Hanczyc and Ikegami (Hanczyc et al., 2007; Hanczyc and Ikegami, 2010) studied a self-moving autonomous droplet. An oil droplet made of oleic acid (about 0.1 mm in size) can move by itself and also react to environmental pH levels. A droplet usually prefers the higher pH regions, depending on some initial conditions and its internal dynamic states. We found that a recent discovery of brain dynamics, the so-called *default network* provides a clear difference between such forms of autonomy (Raichle et al., 2001a). The definition of a default network is the brain activity that is observed while people are in the day-dreaming or resting states. A global (non-periodic) synchrony in neural activity was found to exist in the default mode network.

In this paper, we discuss and characterize the Web autonomy by computing the amount of information flow transferred between different sequences of Google queries as well as Twitter keyword frequencies. Web autonomy is defined as an autonomous active pattern organization without having salient inputs from the real world, which we think can be considered sufficiently close to biological autonomy. Of course, the human activities (e.g., posting and searching keywords) constitute the underlying Web dynamics; however, the human activities themselves are highly controlled by the collective Web pattern (e.g., retweeted posts, Amazon recommendations, Google page ranks, and Web queries). It is like the definition of emergent phenomena in artificial life ((Langton, 1995)) - the causal relationship is not just from

humans (micro) to the Web (macro), but it is from the Web to humans as well. This double causation loop defines the emergent phenomena that we take as evidence of the Web's autonomy. In particular, such Web autonomy has many similar properties with the default mode in the brain as we will discuss in detail later on.

In §2, we review the statistics of Web systems, and we present the method of transfer entropy (TE) as the background for this study. In §3, we give a concrete method of how we analyze the data using TE. In §4, we analyze the data over three months from Twitter and Google over a 3-month period. We then define the TE network and discuss the possibility of the default mode network in the Web. In §5, we discuss the perspectives of autonomy with respect to the default-mode dynamics.

BACKGROUND

Statistics of the Web Systems

It is said that 90% of the Web's data stream was created within the last few years and that the total data amount is getting larger and larger. This exceptional growth is mainly due to emerging social network services (SNSs), such as Twitter and Facebook. It is said that the data volumes are doubling every 2 years, which is even faster than Moore's Law.

The functionality of an SNS was widely recognized after the Egyptian revolution of February 11, 2011, and the Tohoku earthquake of March 11, 2011. Facebook helped bring worldwide attention to the historical event in Egypt. Twitter served as an efficient way for people to communicate and get information on the earthquake. A burst of keywords, such as "tsunami" and "nuclear plant", was observed on and after March 11. SNS and Google react strongly to social movements by producing *burst*-like behavior. This is what we call the *reactive mode* of the Web. Namely, the Web is susceptible to social impacts.

On the other hand, the "normal mode" of the Web can be observed. Even without major social impacts, the Web demonstrates its own dynamics. A constant fluctuation of queries and keyword frequencies with no bursting peaks is observed, which characterizes the normal mode of the Web. We hereafter call this normal mode the *default mode* of the Web. Namely, the default mode provides a baseline activity of the Web, and it may provide a possible mechanism for an artificial system to become autonomous. In the field of Web sciences, an extensive amount of data has been accumulated for the statistics. The famous 6 degrees of separation of the "letter connection" was proposed in 1963 by Milgram (Milgram, 1963) (and more recently by Duncan and his colleagues (Watts and Strogatz, 1998)). Now, by using Twitter, it has been updated to 4 degrees of separation (Kwak et al., 2010). Also, Twitter has some interesting statistics. For example, there are three peaks per day in its number of tweets on weekdays, but this vanishes on the weekends; the time interval between successive tweets obeys the power

law, whose exponent is similar to the rate at which e-mails are received.

Statistics of the memory-related effects on the Web have also been studied by many researchers. One of the early studies shows that the half-life span of crawled Web sites can be approximately 40 to 50 weeks (see (Ntoulas et al., 2004)) and that the ratio of successful downloadable sites is decreased to 80% after the tenth crawl generations. A similar investigation on Twitter has been conducted as well. In the Twitter system, memory is driven by the retweeting of posts, where people repost their favorite tweets on their timelines. Statistics show that the half of the re-tweeting occurs within an hour and 75% in less than a day. However, about 10% of the retweets occur a month later.

When we compare Google and Twitter, we find some unexpected features. Namely, only 126 out of 3,479 unique trending topics (3.6%) from Twitter exist in 4,597 unique hot keywords from Google (Kwak et al., 2010). It is said that those keywords are mostly associated with real-world events, celebrities, and movies. On average, 95 % of topics each day are new in Google, while only 72% of topics are new in Twitter (Kwak et al., 2010). This feature is worth noting, since it reflects that retweet, reply, and mention are prevalent in Twitter, but such interaction among users can never be possible with Google searches. Such interactions might be a factor in ensuring that the same trending topics persist over a relatively longer period of time.

While those statistical properties tell us something about the collective nature of human behavior behind the Web, our interest here is the emergence of Web autonomy; i.e., an intrinsic dynamic of the Web that individual users cannot handle by themselves. For instance, if we take the Web as a living creature, and not as a "slave" machine, what would be the most elegant way to describe the autonomous behavior? Most of the Web's temporal behavior is not stable and periodic; rather, it often shows chaotic, open-ended dynamics. A basic strategy we employ in such a case is to introduce a concept of information flow (Shaw, 1981) that has been developed in the field of nonlinear science.

Entropy Measurements

Physics attempts to take the information-theoretical approach in various fields by extending the concept of entropy. For example, Bennett introduced a notion of logical depth (Bennett, 1988), and Lloyd and Pagels analyzed a thermodynamic depth (Lloyd and Pagels, 1988) to measure the complexity of self-organizing physical processes. However, most of those newly defined entropies are difficult to measure in the pragmatic sense.

More practical applications of the information theory in physics are found in the dynamical systems approach. Among the pioneers of introducing information theory into the dynamical systems, Robert Shaw introduced the notion of information sink and source into the micro-Hamiltonian

systems. He examined the turbulent state as a network flow of sinks and sources of information (Shaw, 1981). Since the late 1980s, many complexity measures have been proposed to characterize chaotic/noisy time series of various kinds. In particular, sequences produced from a chaotic dynamic are indexed with the Lyapunov exponent, fractal dimensions, capacity dimensions, and several information entropies (see, for example, (Ott, 1993)). For example, mutual information is used to study how chaotic instability is linked to noise sources (Matsumoto and Tsuda, 1983).

These information-related entropies have been useful and convenient for detecting the chaotic aperiodicity that has been observed in the experimental sequences, ranging from heartbeats and blood vessel streams to the sun, wind, and optical lasers. However, as it often has been debated, these measures often produce unreliable results, depending on the unknown parameter settings.

In this paper, we compute the information flow of the Web (on Twitter postings and Google queries) based on the TE developed by Schreiber (Schreiber, 2000). Staniek and Lehnertz (Staniek and Lehnertz, 2008; Lizier et al., 2010), and Bertschinger (Bertschinger et al., 2008), TE provides a new information entropy for analyzing a given sequence, particularly, how the future state of the sequence X is determined solely either by its preceding states or by the other sequences. In this sense, TE is similar to the Granger causality (Granger, 1969; Barnett et al., 2009; Ay and Polani, 2008) which calculates the degree to which one sequence drives another. TE, however, offers more advantages than the Granger causality, since when we compare two temporal sequences, TE can remove the false contribution from the common temporal pattern that exists in both sequences. On the other hand, TE cannot measure a causal effect but it rather provides a predictive measure, as was discussed recently (Lizier and Prokopenko, 2010; Chicharro and Ledberg, 2012). Practically speaking, it is more difficult to measure the causal effect without knowing the underlying equation, and also Granger causality is good for linear systems but not so much for highly nonlinear systems. We thus use the TE as for the first step.

By using the computationally feasible quantity called "permutation entropy" (which will be presented in detail in the next section), TE exceptionally differentiates between the upstream and downstream information flow in realistic examples. For example, it has been suggested by (Schreiber, 2000) that TE computes a particular region of the brain that affects other regions in order to help improve the evaluation of patients with epilepsy patients from EEG sequences. Another example is comparing heartbeat and breathing sequences to evaluate which information flows are informationally upstream.

Here, we use the method of TE to characterize the directionality of information flow in sequences of keyword frequency in tweets and Google search queries, and we differ-

entiate between the reactive and default modes on the Web. This approach will provide a useful perspective to understand the Web dynamics in terms of the TE. In the next section, we explain how we compute the TE of the given time sequence in more detail.

APPROACH

Information flow examines how much information is necessary from the rest of the world in order to predict the future state of a system X . Our idea is to define the information flow on the Web and to assign the direction of the information in the Web-state space. If the word "earthquake" is put into Google as a query, for example, it may produce a large number of tweets containing the word "earthquake" in Twitter. In this case, there is an information flow from the Google search query "earthquake" to the tweets containing the same keyword "earthquake".

Transfer Entropy

The usual definition of the Shannon entropy, with the probability distribution $p(x)$, is as follows (where x is an extracted state of a target system; e.g., time sequence X):

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

Using this notation, we define a mutual entropy between two sequences X and Y as follows:

$$MI(X, Y) = H(X) + H(Y) - H(X, Y)$$

By definition, $MI(X, Y) = MI(Y, X)$, so that no causal relationship is detected with MI. By introducing the time delay, we can improve the situation, although it remains difficult to capture the direction of causation. On the other hand, TE from X to Y , which is denoted as $TE(X, Y)$, is defined with the following transition probabilities: $p(x_{t+1}, x_t)$, such as

$$TE(X, Y) = -H(y_{t+1}, x_t, y_t) + H(x_t, y_t) + H(y_{t+1}, y_t) - H(y_t).$$

In the absence of an information flow from X to Y , $TE(X, Y)$ vanishes, as the formula is explicitly non-symmetric with respect to Y and X . Let us express the formulas in a more explicit way such as

$$TE(X, Y) = -\sum p(y_{t+1}, x_t, y_t) \log \frac{p(y_{t+1}|x_t, y_t)}{p(y_{t+1}|y_t)}$$

, where $p(x|y)$ denotes the conditional probability. The opposite effect is obtained in the same manner; for example,

$$TE(Y, X) = \sum_{x_{t+1}, x_t \in X} \sum_{y_t \in Y} p(x_{t+1}, x_t, y_t) \log \frac{p(x_{t+1}|x_t, y_t)}{p(x_{t+1}|x_t)}$$

. We also measure the direction of information flow by comparing the TE for the pair of sequences. In particular, we use

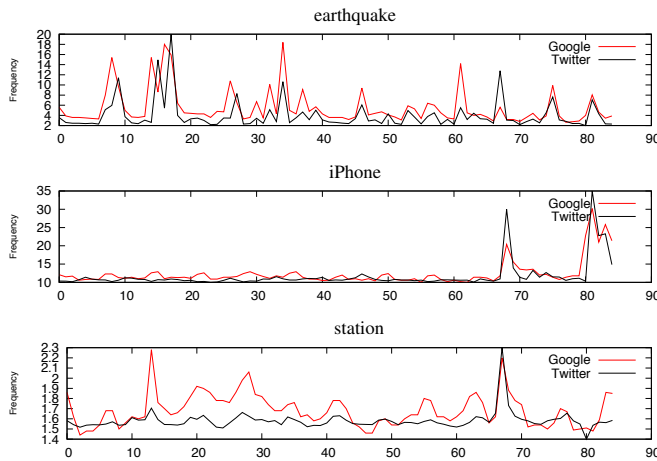


Figure 1: Examples of the sequences of Twitter keyword frequency and Google query frequency from the dataset used for the experiments.

the difference between $TE(X, Y)$ and $TE(Y, X)$ as a quantity of information flow denoted by TE^s through the rest of this paper. In this paper, we apply the technique to the sequences of queries and keywords on Google and Twitter. In order to calculate the TE, we use symbolic sequences rather than the continuous state flow, as it is much more convenient and efficient.

Permutation Entropy

Bandt and Pompe (Bandt and Pompe, 2002) introduced a simple refinement of TE with sequences that are practical feasible coding of the real-world dataset. It is based on the re-ordering of the amplitude values of sequences x_i and y_i , so that the amplitudes are arranged in an ascending order. Namely, $\{(x(n), x(n-1), x(n-2), \dots, x(n-m-1))\}$ are arranged in ascending order and become $\{x(l), x(l+1), \dots, x(l+m-1)\}$ such that $\{x(l) \geq x(l+1) \geq x(l+2), \dots, \geq x(l+m-1)\}$, where m is the embedding dimension (i.e., the effective dimensionality of the target system). We now use the indexes of those variables instead of their amplitudes; for example, in the case of (x_1, x_2, x_3, x_4) , it is re-ordered as (x_4, x_2, x_1, x_3) , so that $(x_4 \geq x_2 \geq x_1 \geq x_3)$ and the new temporal sequence would be $(4, 2, 1, 3)$. Any temporal sequences can be mapped onto one of the $m!$ possible permutations. We use the relative frequency of the symbol sequences and estimate the joint and other probabilities.

Reactive and Default Modes of the Web

A sudden activation of burst in blogspace was analyzed by Kumar et al. (Kumar et al., 2003) by following the hyperlinks of blogs. Gruhl et al. studied how topics propagate through blogspace, and they classified the temporal behavior of the topics by chatters and spikes (Gruhl et al.,

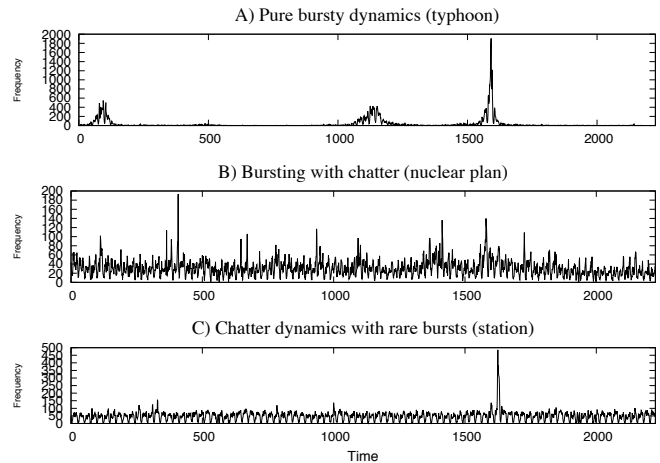


Figure 2: Three types of bursting patterns in a query popularity represented by keywords in Twitter during the study period.

2004). Gruhl et al. looked into blogspace and found that the spikes were mainly triggered by world events, but were rarely caused by the *resonance* within a community. This rare spiking event is a sort of self-organizing effect of the collective motion of users. Gruhl et al. have also characterized and modeled the individual bloggers' networks by using the ideas of infectious disease models.

Our definition of the default and reactive modes of the Web started from the same view (i.e., topics consist of chatters and spikes). We first studied the bursting responses found in the Google search queries and keywords in Twitter. We defined the reactive mode of the Web triggered by the real-world events. We computed the standard deviation of the keyword stream popularity; if the popularity deviated more than the standard deviation, we took it as a burst event. This is also what Gruhl et al. (Gruhl et al., 2004) adopted in their analysis to detect a burst. However, even this simple criterion faces many ambiguous cases; e.g., a large number of large amplitude chatters.

On the other hand, the default mode is a baseline activity of the Web, and our definition of the default mode is about internal synergetic (collective) phenomena. Different from Google queries, people tweet by reading other users tweets, which provides a proverbial "seed" for such cooperative effects. Second, retweets and replies are, in principle, evidence of cooperative phenomena. However, this collective motion cannot be captured simply by the popularity analysis via bursts. We hypothesize that the default mode is a self-persistent activity, so that it is usually buried under the chatter phases. Thus we characterize the sequences using TE to take the information flow into account so as to characterize default and reactive modes on the Web.

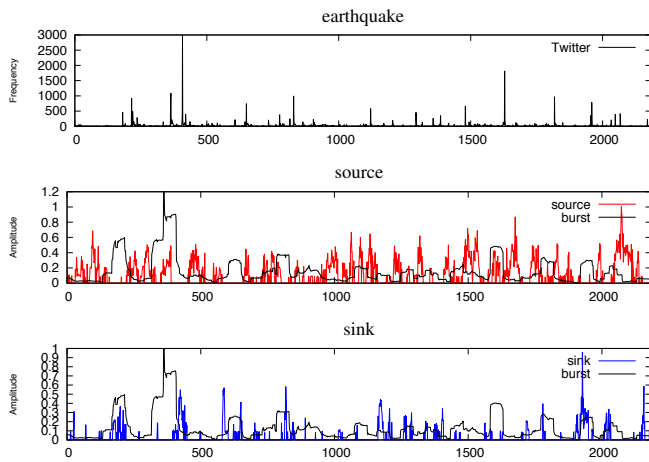


Figure 3: An example of keyword sequences "earthquake." Raw sequence of keyword appearing in Twitter per hour, changes in TE^s of source (in red) and changes in TE^s of sink (in blue) with burst.

We looked into tweets and manually selected 26 keywords (e.g., station, earthquake, tsunami, Steve Jobs, etc.) that cover different popularity dynamics; e.g., with different numbers of bursts and periodicity. For example, when there is an earthquake, people will run a Google search to get information, but will also tweet to communicate with others. These days, we can get information about the earthquake from tweets much faster than from Google search results. Anyway, a large-scale social event typically generates bursts in the time series, and we may also expect synchronous bursts both in the Google and Twitter sequences. Figure 1 lists typical keywords and the associated temporal sequence of keyword frequencies from Twitter and Google. From this figure, we can see that there are many synchronies between Twitter keywords and Google queries. This is apparent in pure-bursting cases such as "iPhone."

Depending on the temporal behavior of the time sequence dynamics, each keyword/query dynamic can be roughly classified into three groups. Similar to Gruhl et al., the following groupings are obtained: (A) pure bursty dynamics, (B) bursting with chatter and (C) chatter dynamics with rare bursts. The representative keyword dynamics for each pattern are depicted in Figure 2. Our analysis on the Google queries on the same 26 set of keywords also showed that the same classification is possible and that the keywords are categorized into the above three categories. However, it is difficult to characterize default and reactive modes via the three burst patterns alone. We thus consider the information flow among keyword/query dynamics to take into consideration the influences on and from other keywords/queries.

To do so, we set the time window (18 days for each keyword) to define and compute the TE of whether the Web

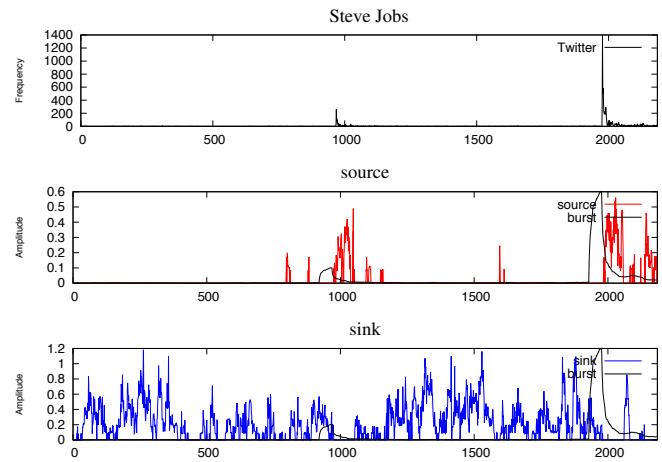


Figure 4: An example of keyword sequences "Steve Jobs." Raw sequence of keyword appearing in Twitter per hour, changes in TE^s of source (in red) and changes in TE^s of sink (in blue) with burst.

state, with respect to the query, is in the default or reactive mode. As we will see in the following sections, some query dynamics show obvious switching from one mode to the other, judging from the classification of the query dynamics. We first computed the TE among sequences of queries/keywords between Google and Twitter. Then we computed the inner information flow among Twitter posts. This is motivated by the fact that people tweet by consciously/unconsciously reading their own timelines, so that the potential content of tweets is connected through local fields (i.e., timelines).

EXPERIMENTS

Data Acquisition

In this study, we picked up a set of meaningful 26 keywords, as well as a randomly selected set of 126 keywords, and for each keyword, we examined the number of queries per day and the number of tweets per day. We collected the query data from Google using Google Trends¹ and Twitter data (only in Japanese) using its APIs for a period of 3 months from July 16, 2001, to October 8, 2011. Figure 1 lists typical keywords and the associated temporal sequence of the frequency of the keywords from Twitter and Google used for our dataset.

Computing Transfer Entropy

TE quantitatively measures the information content of one sequence against the other, and the difference of the TE particularly defines the direction of information flow between two sequences. Here, we compared keyword sequences

¹Google Trends: <http://www.google.com/trends>

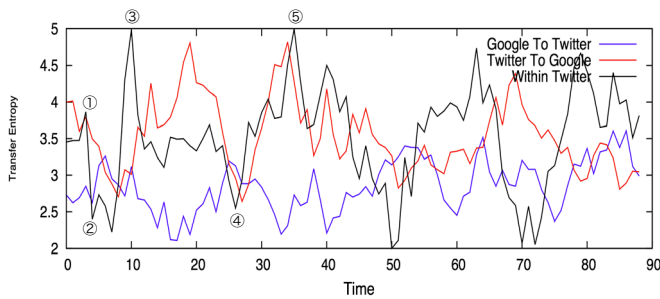


Figure 5: Amount of the sum of transfer entropy flows. (1) Google to Twitter in blue, (2) Twitter to Google in red, and (3) within the Twitter network in black are overlaid in the same figure. A unit of the horizontal axis is per day (the figure is computed for a period of 108 days from July 16, 2011, to October 10, 2011).

within Twitter and between Twitter and Google. Given a pair of sequences, the actual computation consisted of three steps. The first step was to compute an embedded dimension m of the given sequences of a given window size. Here, the window size is defined as 18 steps, which corresponds to 18 days. The second step is to compute the permutation entropy of the m number of the binary sequence space (i.e., the embedded dimension). Here, we evaluated the embedded dimension m as 3 in all sequences. Finally, the thrust step is to compute the TE of each window. We shift the window by one step and then repeat all three steps. The window size and the embedded dimension are varied to check the reliability of the computed transfer entropy. The minimal window sized is limited mainly due to the Google API. The embedded dimension is tested from one to eight, but we did not see any significant improvements above three.

For each pair (i, j) of nodes (i.e., query/keyword), the difference of the TEs (i.e., $TE(i, j)$ and $TE(j, i)$) where generally $TE(i, j) \neq TE(j, i)$ computes the direction of the information flow. We have computed all of the TEs for all of the pairs of queries from Google searches and keywords from Twitter.

Transfer Entropy Network

First of all, we studied the meaningful 26 keywords and $TE^s(i, j)$ is computed for every keywords pair (i, j) . Using the $TE^s(i, j)$ as a distance matrix between the keywords (i, j) with an adequate threshold value (th), we draw a transfer entropy network (i.e., each node of the network is a keyword, and a pair of nodes are connected if the TE^s is greater than th).

In the following, we explain how we classify three kinds of nodes in a TE network: sink, source and others. A sink node has only incoming flows, and a source node has only outgoing flows. Then, the sum of TE^s for those sink and source nodes will be used as the nodes in a TE network.

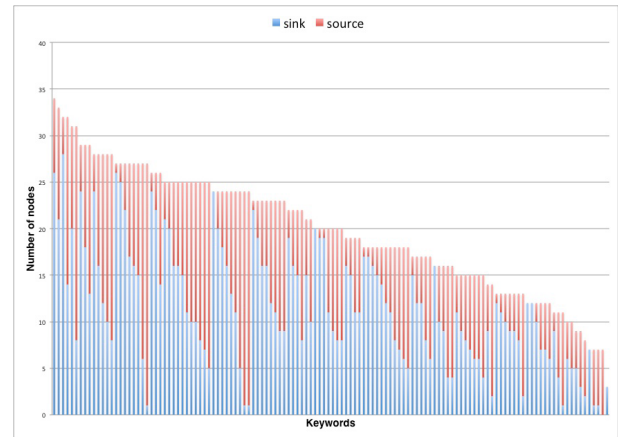


Figure 6: A role of keywords (sink=blue, source=red) during the period is counted and plotted in the descending order (of the total number of sink and source). Some keywords always behave as the sink or source, while others change from one to the other for the duration of the experiment.

Figure 3 and Figure 4 show the raw data sequence and the sum of TE^s of source nodes (in red) and sum of TE^s on sink nodes (in blue) with a significant bursting pattern superimposed. Here, a significant burst is defined when the amplitude exceeds a $\sigma + \mu$ (where σ is a standard deviation and μ is a mean value). As depicted in Figure 3, the keyword "earthquake" has many bursts that are sometimes synchronized with TE^s of the sink nodes (blue) and sometimes that with the source nodes (red). The keyword "Steve Jobs" has many bursts of sink TE^s , as shown in Figure 4. A TE^s of a source node nicely synchronizes with the raw population of bursts, and when there is no burst, TE^s , as a sink becomes very active.

Suppose that Google queries are relatively more sensitive to the real world; that is, information flow from Google and Twitter measures how Google queries affect the Twitter community. From the "Steve Jobs" example, we hypothesize that (1) a burst event is followed by the burst of TE of a source nodes (red), that (2) chatter states are defined by no bursting events in the popularity of keywords, and that (3) the TE of the sink nodes only show bursting behavior. But the example of "earthquake" does not always follow this pattern. Therefore, as the next step, we used 126 randomly selected keywords, apply the same analysis, and, on top of it, compute the internal TE^s cluster changes, its size, and the amount of flow.

Dynamics of the Transfer Entropy Network

Using 126 random keywords, we study the behavior of inner-TE flow networks on Twitter as defined above. A TE network changes its size and connectivity, which are correlated with the incoming and outgoing information flows

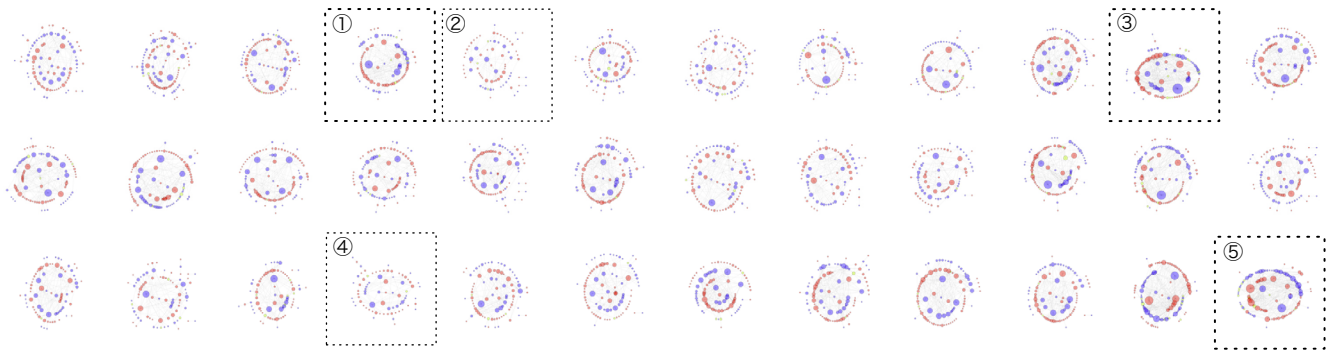


Figure 7: A snapshot of the transfer entropy network within Twitter sequences over time is computed. A red-colored node is the source, a blue-colored node is the sink, and the size of a circle corresponds to the amplitude of TE^s of each node. A snapshot from different time steps is picked up and numbered from 1 to 5, which corresponds to the numbers in Figure 5. It should be remarked that the density of connection gets higher in numbers 1, 3, and 5, where the amount of flow attains maximum, in Figure 5, while the density gets lower in numbers 2 and 4, where the amount of flow attains minimum.

between Google and Twitter. We illustrate how the largest connected network significantly varies its size over time in Figure 5.

Figure 6 shows the changes in the number of sink and source nodes over time for each keyword, and Figure 7 illustrates the TE network depicted by assigning an edge to the difference, TE^s , whose value is larger than $th = 0.15$. The temporal reconnection of the networks in Figure 7 represents the temporal variation of information flow among keywords. The following two points will be made from this network analysis.

- (i) Some keywords always behave as sink or source, while others change from one to the other for the duration of the experiment.
- (ii) The size of the inter-TE flow networks on the Twitter network increases when the incoming flow from Google to Twitter decreases, and the size decreases when the incoming flow increases. Also, the total amount of TE increases (decreases) according to the decrease (increase) of the incoming flow between Google and Twitter.

In accordance with the notion of the default mode network of the brain, the default mode of the Web can be primarily characterized by the amount of incoming information flow (or sink), complementary to the mass of bursts. This is because, like the default mode in the brain system, the inner TE^s network becomes suppressed when there is incoming flow, which we assume is connected to significant real events, and becomes activated when there is more outgoing flow (i.e., a "resting" state). The network pattern in Figure 7 has some keywords common to the temporary varying network over a long period of time as seen in Figure 6. The nodes with a fixed role (i.e., sink or source) that we found in this analysis are the candidates for the element of the default mode.

Of course, the number of keywords analyzed in our experiment (i.e. 126 keywords) is a very small portion of the entire Twitter keyword set. Nevertheless, we hypothesize that the existence of such information flow networks can be a core engine for sustaining Web autonomy. We are now analyzing much larger sized networks, which will be reported elsewhere.

DISCUSSIONS

This paper has explored how to refine the reactive mode and default mode on the Web and how to determine the current drawbacks associated with understanding the dynamics of the Web. As a result, we found that Google query sequences and Twitter keyword sequences mutually affect each other. Characterizing the information flows of the sequences was very successful using TE. To our great surprise, different keyword sequences from Twitter also mutually affect each other, and we observed a strongly connected network of the set - comprised of Twitter keywords - that changes its size and connection strength. The original idea of the reactive and default modes came from brain science (Raichle and Snyder, 2007; Raichle et al., 2001b). A brain region responsible for a given task is identified by measuring the neural activity that is observably higher compared to a baseline activity. A natural question is posed by Raichle et al.: What is the baseline neural activity and how we can measure it? They studied baseline activity by analyzing the regions that become less active when a specific task is given. This successful approach uncovered some remarkable perspectives about the default mode: (1) the area associated with the default mode is found in the parietal association area of the posterior Cingulate gyrus; (2) the neural activity of that area becomes suppressed when there is a specific task, which is how the default mode has been identified; (3) there is global synchrony among these brain areas; (4) the default mode has

something to do with the creative capability of a brain system; and (5) the area of default mode is found where the episodic memory is believed to be processed (see, for example, (Sestieri et al., 2011)).

Our definition of and findings related to the default mode in the Web can be discussed in a similar manner to the default mode in a human brain system. Differentiating between these two modes, the reactive and default, will provide a useful perspective toward understanding the Web dynamics and predicting the future of bursting behavior in sequences of keyword frequencies in tweets, as well as sequences of search queries in search engines like Google. Before the analysis described in this paper, we had two assumptions to characterize the default mode of the Web:

- a) a bursting state without having any relation to significant real-world events, and
- b) a baseline activity of the Web without having apparent bursting behaviors.

Our analysis of the data revealed that

- c) the autonomous oscillating behavior (of its characteristic periodicity found around a few weeks) observed in the TE network among Twitter sequences is a candidate for detecting the Web default mode, and
- d) when the TE network grows, the outward information flow from Twitter to Google increases, which can be taken as a spontaneous activity of the Web with respect to Twitter and Google.

Based on the observation, we turn down the property (a) and modify (b) as a baseline activity that sometimes produces bursting behavior spontaneously. We are now investigating the properties (c) and (d) with a larger dataset, and more convincing results will be reported elsewhere.

Concerning the examples of artificial life systems, we think that the default mode is a key issue for artificial life studies (Ikegami, 2012). Oil droplets and other artificial life systems (e.g., robots or autonomous sensing systems) possess primitive forms of the default mode with different time scales. Instead of simply saying that artificial life is autonomous if it is driven by its own program (e.g. computer viruses), it would be more fruitful to seek for the conditions of the potential default mode such as we raised in this paper. By finding the default mode, we can bridge the gap between biological autonomy and autonomy of computer programs or that of chemical oil droplets.

Acknowledgement

We thank Yasuhiro Hashimoto and Eiko Matsuda for their substantial support regarding data analysis. This work was supported by the JSPS Grant-in-Aid for Young Scientists (B) (#23700106 "Research and Development of space design supporting tool using Web information"), partially by

the JSPS Grant-in-Aid for Scientific Research (B) ("Towards the Construction of Technological Philosophical Extension of Ecological Phenomenology"), by JSPS Grant-in-Aid for Challenging Exploratory Research ("Development of the Revolutionary Experiment Setups for Studying Artificial Life Systems"), and by Grant-in-Aid for Scientific Research on Innovative Areas ("The study on the neural dynamics for understanding communication in terms of complex hetero systems").

References

- Ay, N. and Polani, D. (2008). Information flows in causal networks. *Advances in Complex Systems*, 11:17–41.
- Bandt, C. and Pompe, B. (2002). Permutation entropy - a complexity measure for time series. *Phys. Rev. Lett.*, 88:174102.
- Barnett, L., Barrett, A. B., and Seth, A. K. (2009). Granger causality and transfer entropy are equivalent for gaussian variables. *Phys. Rev. Lett.*, 103:238701.
- Bennett, C. H. (1988). *Logical Depth and Physical Complexity in The Universal Turing Machine- a Half-Century Survey* (pp.227-257). Oxford University Press.
- Bertschinger, N., Olbrich, E., Ay, N., and Jost, J. (2008). Autonomy: An information theoretic perspective. *Biosystems*, 91:331–345.
- Chicharro, D. and Ledberg, A. (2012). When two become one: The limits of causality analysis of brain dynamics. *PLoS ONE*, 7:e32466.
- Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37:424–438.
- Gruhl, D., Guha, R., Liben-Nowell, D., and Tomkins, A. (2004). Information diffusion through blogspace. In *Proceedings of the 13th international conference on World Wide Web*, pages 491–501.
- Hanczyc, M. M. and Ikegami, T. (2010). Chemical basis for minimal cognition. *Artificial Life*, 16(3):233–243.
- Hanczyc, M. M., Toyota, T., Ikegami, T., Packard, N., and Sugawara, T. (2007). Chemistry at the oil-water interface: Self-propelled oil droplets. *J. Am. Chem. Soc.*, 129(30):9386–9391.
- Ikegami, T. (2007). Simulating active perception and mental imagery with embodied chaotic itinerancy. *J. Consciousness Studies*, 14:111–125.
- Ikegami, T. (2012). A design for living technology: Experiments with the mind time machine. *Artificial Life*, in press:–.
- Kumar, R., Novak, J., Raghavan, P., and Tomkins, A. (2003). On the bursty evolution of blogspace. In *Proceedings of the 12th international conference on World Wide Web*, pages 568–576.
- Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is twitter, a social network or a news media? In *Proc. of the 19th International World Wide Web*, pages 591–600.
- Langton, C. G. (1995). *Artificial Life: An Overview*. MIT.

- Lizier, J. T., Heinzle, J., Horstmann, A., Haynes, J.-D., and Prokopenko, M. (2010). Multivariate information-theoretic measures reveal directed information structure and task relevant changes in fmri connectivity. *Journal of Computational Neuroscience*, 30:85–107.
- Lizier, J. T. and Prokopenko, M. (2010). Differentiating information transfer and causal effect. *European Physical Journal B*, 73:605–615.
- Lloyd, S. and Pagels, H. (1988). Complexity as thermodynamic depth. *Annals of Physics*, 188:186–213.
- Matsumoto, K. and Tsuda, I. (1983). Noise-induced order. *Journal of Statistical Physics*, 31:87–106.
- Milgram, S. (1963). Behavioral study of obedience. *Journal of Abnormal and Social Psychology*, 67:371–8.
- Ntoulas, A., Cho, J., and Olston, C. (2004). What’s new on the web? the evolution of the web from a search engine perspective. In *Proc. of the 13th International World Wide Web*, pages 1–12.
- Ott, E. (1993). *Chaos in Dynamical Systems*. Cambridge University Press.
- Raichle, M. E., MacLeod, A. M., Snyder, A. Z., Powers, W. J., Gusnard, D. A., and Shulman, G. L. (2001a). Inaugural article: A default mode of brain function. *PNAS*, 98:676–82.
- Raichle, M. E., MacLeod, A. M., Snyder, A. Z., Powers, W. J., Gusnard, D. A., and Shulman, G. L. (2001b). Inaugural article: A default mode of brain function. In *Proc. of the National Academy of Sciences*, pages 676–82.
- Raichle, M. E. and Snyder, A. Z. (2007). A default mode of brain function: A brief history of an evolving idea. *NeuroImage*, 37(4):1083–1090.
- Schreiber, T. (2000). Measuring information transfer. *Phys. Rev. Lett.*, 85:461–464.
- Sestieri, C., Corbetta, M., Romani, G. L., and Shulman, G. L. (2011). Episodic memory retrieval, parietal cortex, and the default mode network: Functional and topographic analyses. *The Journal of Neuroscience*, 12:4407–4420.
- Shaw, R. (1981). Strange attractors, chaotic behaviour and information flow. *Zeitschrift fur Naturforschung*, 36A:80–112.
- Staniek, M. and Lehnertz, K. (2008). Symbolic transfer entropy. *Phys. Rev. Lett.*, 100:158101.
- Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *Nature*, 393:440–442.

Evolution of a Communication Code in Cooperative Tasks

Aditya Rawal¹, Padmini Rajagopalan¹, Risto Miikkulainen¹ and Kay Holekamp²

¹The University of Texas at Austin

²Michigan State University

aditya@cs.utexas.edu

Abstract

Communication through vocalizations is used by spotted hyenas and chimpanzees for coordination during hunting and for raising alarm calls in defense (Bullinger et al., 2011; Holekamp et al., 2007). Vocal signals are omni-directional and are therefore more effective than visual communication in these situations. In cooperative tasks, agents use these signals to pro-actively exchange information for common good. A simulated predator-prey domain is considered in this paper - where multiple predator agents exchange real valued messages as an approximation of vocalization in nature. In artificial intelligence, the problem of coordination among multiple predator agents during prey capture is hard because of the non-Markovian environment (Panait and Luke, 2005). Experiments are carried out in this paper to show how information exchange through messaging can make the environment less non-Markovian and improve predator team performance during cooperative hunt. The values of these messages are analyzed to study the emergence of a common communication code among the predator agents. The results in this paper also provide an insight into the constraints under which language evolves in nature.

Introduction

Spotted hyenas employ vocal signals for kin-recognition (Holekamp et al., 2007), chimpanzees use it for building a consensus before embarking on a group hunt (Bullinger et al., 2011) and both hyenas and vervet monkeys (Vaughan et al., 2011) raise vocal alarm calls when under predatory threat. In nature, communication through vocalization plays an important and diverse role especially in scenarios where coordination among group of individuals is required. Coordination in teams translate into multi-agent problems in artificial intelligence (AI) domain. As in the real world, not all the information about each state in the simulated environment is known, so the agents cannot consistently select the optimal action. From the AI perspective, this non-markovian nature of the environment in multi-agent problems is a major challenge in building cooperative teams. Animals alleviate this problem of group coordination through constant and pro-active transfer of information among concerned individuals through various forms of communication - like visual, vocal, tactile

and olfactory. One advantage of vocal signalling over other modalities of communication is the fact that it is omni-directional and can travel long distances. Another interesting characteristic of vocal language is that it is usually consistent among the members of a species. Infants learn this language while growing up, and begin actively participating in societal roles like hunting and protecting.

Inspired by such instances in nature, we simulate teams of cooperative predator agents in prey capture tasks, with the goal of evolving a common communication code among them. As an approximation to the vocal signalling in nature, the predators are provided with continuous channels through which they can send and receive real-valued messages/codes among themselves. Language usually has two aspects to it - conveying meaningful information on the part of the sender and ability of the receiver to interpret this information. The first two experiments in this paper study the constraints under which such a common predator code emerges. The third experiment compares the performance of real-valued messaging with direct communication (an approximation to vision in nature). The predator agents are evolved using Multi-component ESP (a neuroevolution technique), which has previously been found to be successful in such sequential decision making (for prey capture) tasks (Rawal et al., 2010).

Background and Related Work

One of the first simulations in artificial organisms to study the emergence of language was done by (Werner and Dyer, 1992). They used discrete signals to evolve a communication protocol among agents for the task of mate selection. (Saunders and Pollack, 1996) applied both discrete and continuous for food search task and also analyzed the different evolved signals among agents. More recently, (Tuci and Vicentini, 2007) conducted an experiment in a team of 3 robots, where each robot is equipped with different sensors. With limited perception of the world, the robots are forced to cooperate by communicating sensory information among each other. A single controller is cloned and is used for controlling all the three robots in their experiments. (Jim and

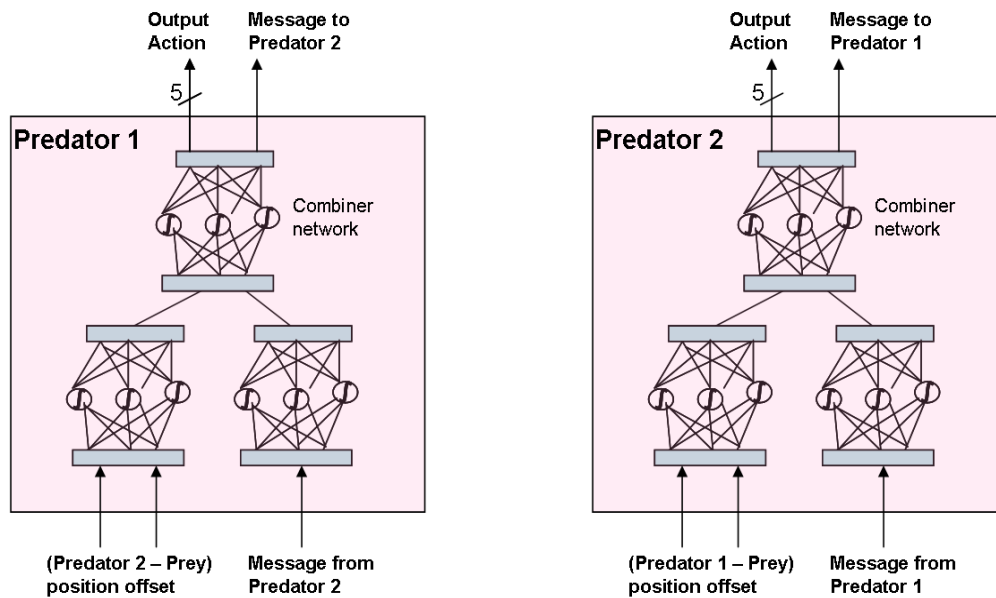


Figure 1: Experiment 1 - Predator agent architecture. Each agent controller is constructed using Multi-component ESP (Rawal et al., 2010). Each predator agent controller has two input sensory neural networks - one for tracking the position offset between the prey and the other predator and second for sensing the real-valued message from the other predator. The outputs of the two input neural networks are combined using a combiner network. The combiner network has 5 output nodes corresponding to 5 possible predator action, and one node for the output message to be sent to the other predator.

Giles, 2001) have used predator-prey domain to show how communicating agents with evolved signalling outperform non-communicating agents. (Knoester et al., 2007) evolved artificial organisms for distributed problem solving through communication. Their experiment demonstrated how information propagates in multi-agent settings.

This paper takes a different approach to study the evolution of communication in artificial agents. It first aims to study the situations under which a common communication code emerges through messaging among a team of evolving predators. We believe that the knowledge of such constraints would help in understanding the evolution of communication in nature. Second, it compares the efficacy of such a team built using messaging as compared to direct communication. Similar to (Saunders and Pollack, 1996), we use real-valued communication channels for signalling among predators.

The predator-prey domain used as a testbed in this paper is a special case of the pursuit-evasion domain. There are predators and prey on the field at the same time and the predators have to capture the prey while the prey try to evade the predators. In these experiments, a team of predators is evolved using cooperative coevolution to capture the prey. The world in this simulation is a discrete toroidal environment with 100 x 100 grid locations without obstacles, where the predators can move in four directions: east, west, north and south. They move one step at a time, and all the agents

take a step simultaneously. To move diagonally, an agent has to take two steps (one in the east-west direction and one in the north-south direction). A predator is said to have caught a prey if it moves into the same location in the world as the prey.

Multi-component ESP (Rawal et al., 2010), a hierarchical cooperative coevolution architecture is used to construct separate controllers for each predator agent. This neuroevolution architecture has previously been successfully used to coevolve a team of predators hunting prey (Rawal et al., 2010; Rajagopalan et al., 2011). It allows for a single agent controller to be composed of multiple networks - where the networks cooperate and their outputs are combined using a combiner network (figure 1). Networks within a controller can be dedicated to different subtasks that the agent must carry out, or for tracking different pieces of information it senses from the environment. Each of these networks is composed of neurons, which represent connection weights for a given node in the network. Each of these neuron is evolved separately in a subpopulation of its own. The final fitness is obtained by constructing a network out of neurons picked randomly from their respective subpopulations and evaluating it in the domain. The fitness received by the network is then assigned to its component neurons. If such a process is carried out several times, selecting neurons at random, each individual neuron's fitness, calculated by averaging over the number of times it was picked, gives a rough

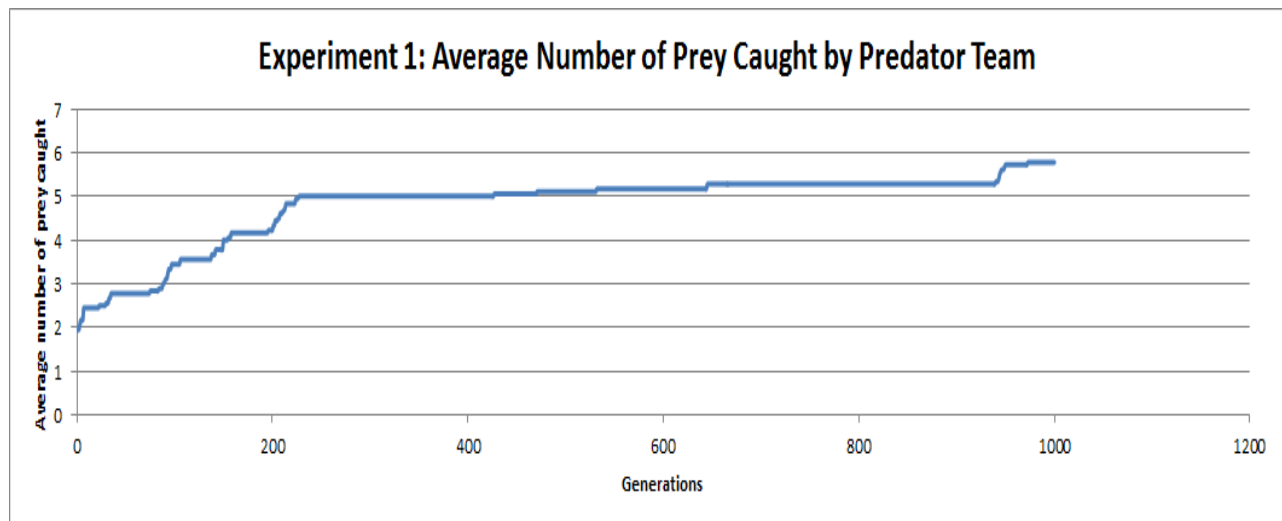


Figure 2: The average number of prey caught by the predator team at every generation in experiment 1. The predators switch between two roles - being 'blind' and mobile in one and stationary (with vision) in the other. In order to successfully capture the prey, the stationary predator should guide the 'blind' mobile predator by sending meaningful messages. The agents thus evolve communication for sustaining cooperation. The results shown are average of 10 runs.

indication of how good that component of the neural network is. This approach breaks up the task into manageable subtasks, thus making the search space smaller, and avoids competing conventions among the neurons.

All the agents on the field share rewards obtained. Reward sharing has been shown to be very effective as an incentive to evolve cooperation among agents (Yong and Miikkulainen, 2009; Rajagopalan et al., 2011). and therefore used here as well.

Experimental Setup

All the predators on the field are evolved using the Multi-Component ESP architecture (Rawal et al., 2010). Three experiments are performed in this paper for prey capture tasks. The goal of the first two experiments is to study the emergence of a communication code among the predators. The third experiment is designed to highlight the utility of this code during cooperative hunting tasks. In all the three experiments, there is a single non-evolving prey in the world. The prey re-appears randomly at a new location once it gets caught. This allows the predators to capture several prey in a single trial/episode. The prey is stationary in the first two experiments, while it moves with a speed of 0.75x in the third experiment. The environment employed is the 100x100 toroidal grid world first used successfully in the predator-prey domain in (Yong and Miikkulainen, 2009). The predators consist of one or more input sensory networks for tracking each bit (either real-valued message or discrete position offsets) of information sensed from the environment. The predators with more than one input networks also include a combiner network which combines the output of

these networks to generate the next predator action and/or predator message. All the predator networks are triggered at every time step of the episode. Each network (including the combiner) has a feedforward architecture with a single layer of 10 hidden neurons and sigmoidal activation functions. Each hidden neuron is evolved in a separate subpopulation consisting of 100 neurons; each neuron is represented as a concatenation of real-valued numbers representing full input and output connection weights.

At the beginning of each generation, 1,000 trials are conducted, and for every trial, a set of neurons is chosen at random from the subpopulations to construct the predators. Each such unique team of predators is evaluated in five simulation runs. All the predators move synchronously, taking one step at each timestep. There are five different actions possible: move up, down, left or right, or remain idle. Each simulation run consists of 500 timesteps in the first experiment, and 300 timesteps in the second and third experiments, during which the predators attempt to catch prey. Each prey gives a reward of 100 points on capture, which is shared equally among the predators. Reward sharing has previously been shown to be effective in fostering the evolution of cooperation in predators (Rajagopalan et al., 2011). The fitness obtained from averaging the total rewards earned over the five runs is then assigned to all the neurons that were used in building these predators.

After the trials, the top 25% of neurons within each hidden neuron subpopulation are selected for recombination. A chromosome is a string of real valued weights associated with each hidden neuron. Since the gene-length of chromosome is fixed, the recombination involves blending real

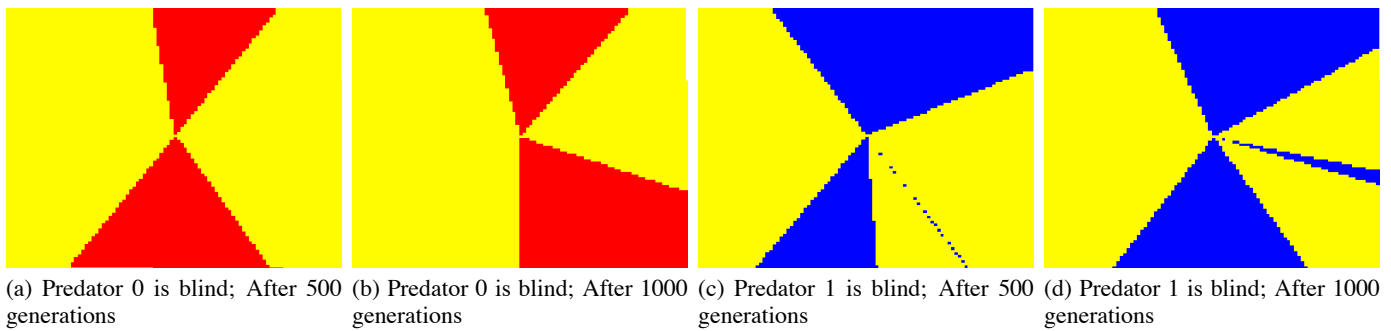


Figure 3: Experiment 1: The best predator team after several generations of evolution is picked for this plot. Each grid cell is painted a color based on the action of the blind predator when present in that cell. The other predator is stationary and fixed at (0,0) and sends messages to the blind predator based upon which it takes action. The prey is also fixed at (50,50). The color coding is as follows- red denotes go down, green denotes go right, blue denotes go up, yellow denotes go left, and black is remain idle. Each agent has evolved to send and receive only two commands - "go down" and "go left" or "go up" and "go left".

valued weights from the same position in the gene using simulated binary crossover (Agrawal and Deb, 1994). The offspring replace the bottom 50% of the neurons in the corresponding subpopulation. Mutation is carried out with a probability of 0.4 on one randomly-chosen weight on each chromosome, by adding a Cauchy-distributed random value to it. Small changes to these parameters lead to similar results.

Experiment 1

An experiment with two predators and a prey was conducted to study the evolution of a common code among predators during hunting. As shown in figure 1, each predator agent has two sensory input networks - one for sensing the offset between the other predator and prey and second to sense the message from the other predator. In order to simplify the analysis of the results, the predators switch between two states - being 'blind' (with no information about prey and other predator position) and being stationary (fixed position). For example, in the first half of an episode, predator 1 is blind i.e it receives no input in its sensory network dedicated for tracking offset between predator 2 and the prey. However, with the messages it receive at every time step from the other predator, it can make decisions to move in the world. During the same time (first half of episode), predator 2 is stationary but it can track the position offset between predator 1 and prey. These predator roles are switched in the second half of the episode. The predators are therefore required to cooperate to successfully catch the prey. One way of doing this is to perfect the system of communication by evolving a code where different real-valued messages represent different pieces of information or commands. The swapping of roles also plays an important part in the evolution of a messaging code. It ensures that both predators evolve not only the ability to send a message in the evolved

code, but also to interpret incoming messages correctly and take subsequent action.

The results of this experiment are given in figures 2 and 3. Figure 2 is a graph of the average number of prey caught by the predator team in each generation. It can be seen that the two predators evolved to catch more than one prey in every episode. Figure 3 was generated by setting one of the evolved predators at location (0,0), and the prey at location (50,50) and setting the blind predator at all the grid cells in the world. The color of any cell in the diagram represents the action taken by the blind predator at that location in the world after receiving a message from the other (stationary) predator. Here, red represents the action "go down", green represents "go right", blue is "go up", yellow is "go left", and black is "remain idle". From figure 3, it can be seen that the blind predators always take one of the two actions - "go right" or "go left", and "go up" or "go down". These actions correspond to the two fixed real values that the stationary predators have evolved to send (not shown here). Evolution has discovered that two commands are enough for prey capture. Although both predators evolved the ability to send and interpret meaningful messages, their communication code is not consistent.

Experiment 2

We have seen from the previous experiment that two predators are capable of evolving a messaging code to communicate useful information between themselves. But the messages sent by the first predator and received by the second may be in a different code from the messages sent by the second predator to be received by the first. It is a logical next step to investigate the circumstances under which different agents can evolve a common code for communication.

To this end, this second experiment was devised, where three predators coevolve to catch prey (see figure 4). In

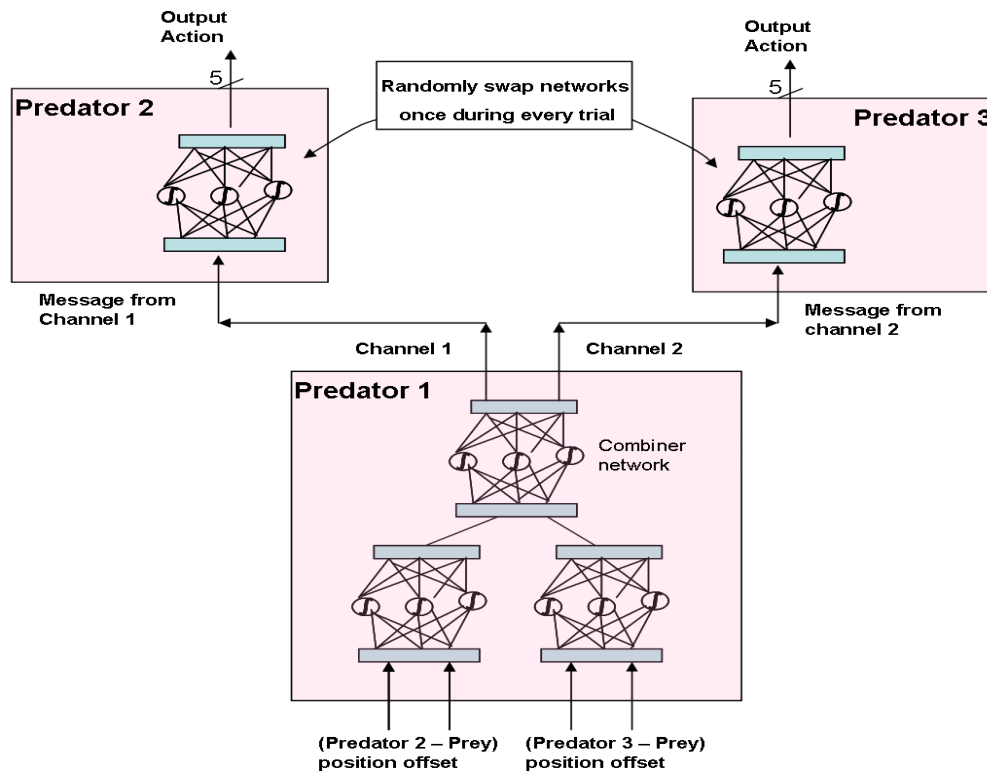


Figure 4: Experiment 2 - Predator agent architecture. Each agent controller is constructed using Multi-component ESP. Predator 1 agent controller has two input sensory neural networks - each for tracking the position offset between the prey and predator 2 and predator 3 respectively. Predator 1 is stationary and has two output channels for sending real-valued messages. Predator 2 and 3 are mobile and 'blind' and by interpreting the messages from predator 1. In order to facilitate evolution of a common communication code between the predators, the neural network controllers of predators 2 and 3 are switched once in every episode.

this case, predator 1 is always stationary and can see the prey and as well as the other two predators. The other two predators (predators 2 and 3) are blind and mobile, receiving real-valued messages from predator 1 to determine their movement. Predator 1 has two output channels of communication, that is, two output nodes to evolve two different messages to send to the two predators. In order to facilitate the two channels to evolve common message codes, a trick is used. The controllers (i.e. the neural networks) of predator 2 and 3 are swapped once in a while so that each network starts receiving messages from the other channel. Since each output communication channel of predator 1 is associated with a particular predator body and location, it will now be sending messages to the either of the two controllers.

This swapping of the neural networks in blind predators will ensure that a single common messaging code has to be evolved for any successful prey capture to occur. This is because each network must evolve to interpret the messages from both channels correctly without knowing which channel it is receiving the message from. Therefore, it evolves to treat messages from both sources as coming from the same

communication protocol. In turn, the stationary predator has to send a meaningful message from each of its channels without knowing how the receiving predator will interpret it, because it does not know which network is receiving it. Thus it will evolve to send messages in the same code from both its channels.

The results of this experiment are in figures 5 and 6. The graph in figure 5 shows that the predator team is successful in catching more than one prey on average in each run. As in the previous experiment, figures 6 were created by placing the evolved predators at certain locations in the grid world and recording the actions they take at each grid cell. The stationary predator was placed at location (0,0) and the prey at location (50,50). Each blind predator was placed at all the locations in the world and its output actions were represented by a color for that cell. As before, red denotes "go down", green denotes "go right", blue denotes "go up", yellow denotes "go left", and black is "remain idle". The input to the predator was a message from one of the two communication channels of the stationary predator. Figure 6a represents the actions of blind predator 1 as a response to

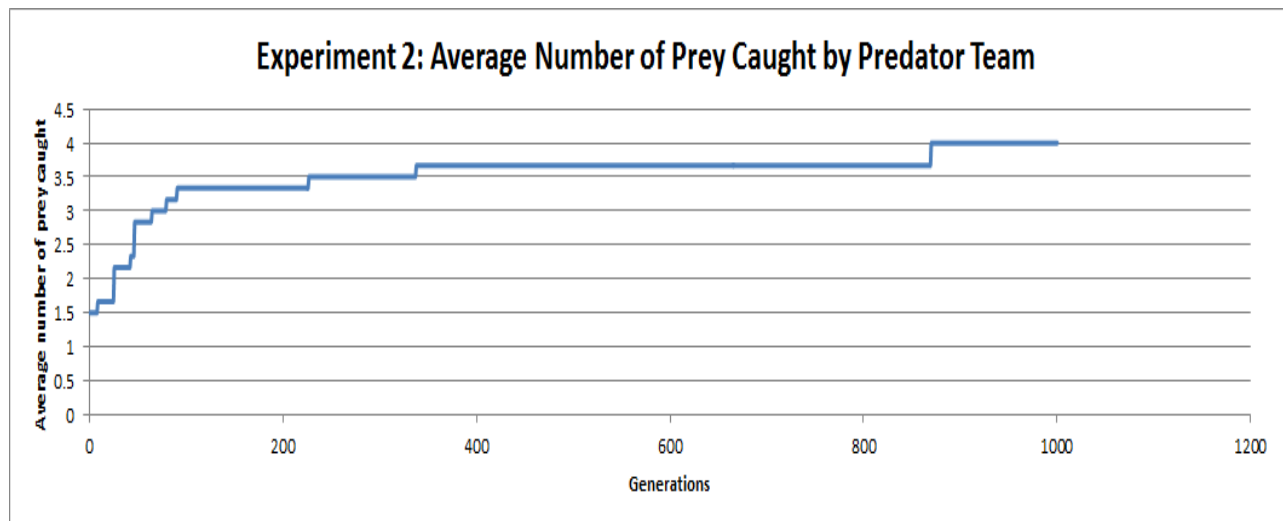


Figure 5: The average number of prey caught by the predator team at every generation in experiment 2. There are two 'blind', mobile predators and one stationary predator with vision. The controllers of the mobile predators are switched once in every episode. The predator has evolved to successfully accomplish the prey capture task.

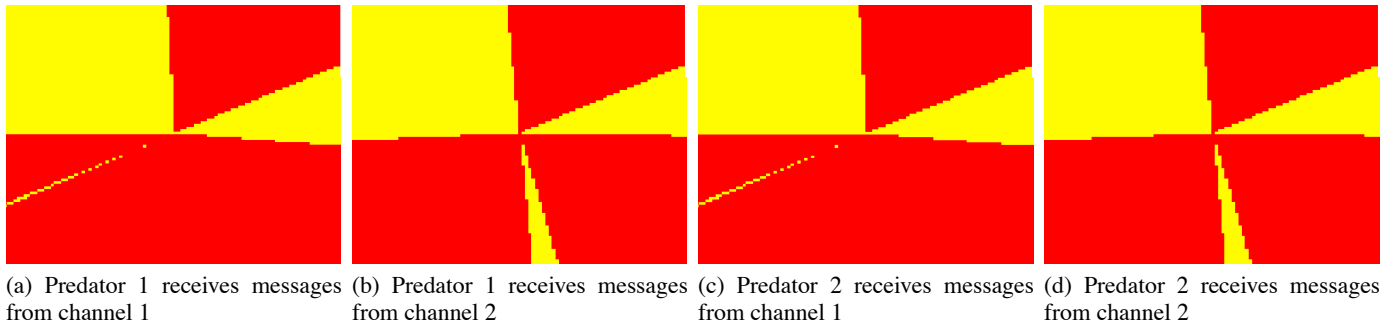


Figure 6: Experiment 2: Each grid cell is painted a color based on the action of the blind predator when in that cell after many generations of evolution. The communicating predator is stationary and fixed at (0,0) and sends messages to the blind predator. The other blind predator is not on the field. The prey is at (50,50). The color coding is same as described in earlier.

communication channel 1, figure 6b corresponds to predator 1 and channel 2, figure 6c corresponds to predator 2 and channel 1, and figure 6d to predator 2 and channel 2. Predator 1 response (action) to channel 1 message is the same as predator 2 response for channel 1. Similarly, the response of both the predators is the same for channel 2 as well.

Experiment 3

After demonstrating the evolution of a common communication code among predators in previous experiments, we now assess its utility. Two experiments, comparing messaging and direct communication, are performed here. Direct communication is analogous to vision in nature - where the predators can observe each other's position. The agent architecture used in both messaging and direct communication is same as that of experiment 1 (Figure 1), however there are changes in the inputs. For messaging, the first

input sensory network of predator 1 tracks the prey position and its second sensory network receives messages from predator 2. Similarly, the two input networks of predator 2 tracks prey position and receives messages from predator 1. For direct communication, each predator tracks the prey position (input network 1) and other predator position (input network 2). The task is made more challenging as the prey now moves in the world (prey speed = 0.75x predator speed) The prey follows a fixed policy of moving away from the nearest predator. The performance comparison of messaging and direct communication is shown in Figure 7. Messaging between predators results in slightly better performance than direct communication. Although the agents evolve a few commands during messaging, these codes are simple and flexible enough for prey capture. The videos for the evolved predator behaviors can be found at: nn.cs.utexas.edu/?alife_2012_communication

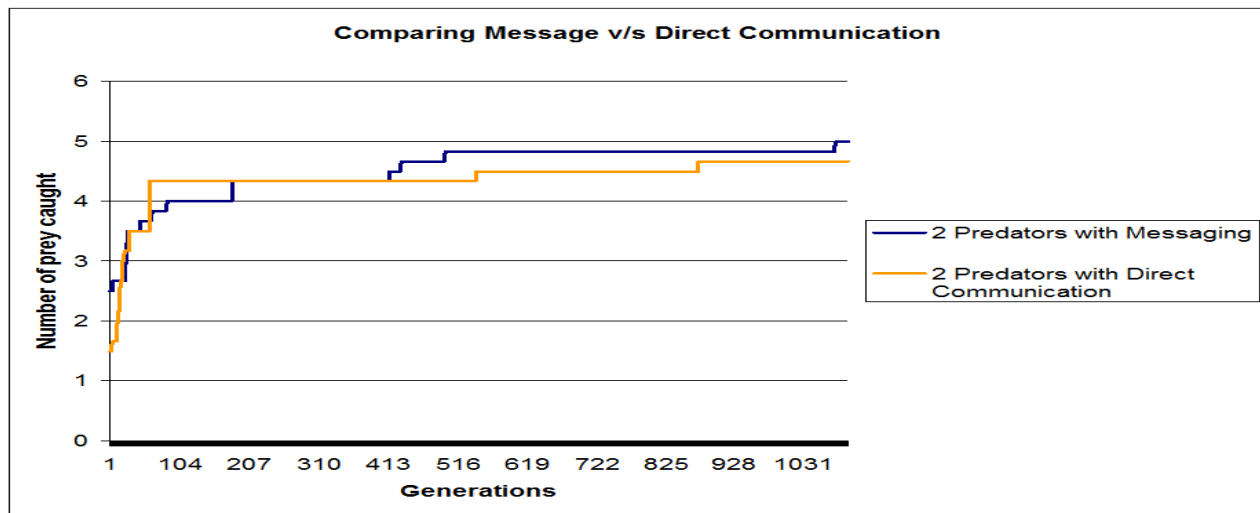


Figure 7: Performance comparison between messaging and direct communication. Two predators cooperate to capture a mobile prey (prey speed = 0.75x predator speed). The agent architecture is similar to the one in Figure 1. In messaging, agents can sense the prey as well as receive real valued signals from each other. In direct communication, the agents sense the prey and each other's position. The results are average of 5 runs. Predators in messaging set-up perform slightly better than direct communication.

Discussion and Future Work

The results of the first experiment (figure 2) demonstrate that the two predators successfully evolved to catch prey. As established in the previous section, one of them was blind and the other was stationary at every point during the simulation. Thus the stationary predator had to evolve a messaging code to communicate its knowledge to the blind predator. And since the two predators exchange their roles during the simulation run, both predators evolve the ability to both send meaningful messages and interpret them correctly. This can be seen more clearly from figure 3, which show the actions taken by the blind predator in response to a message from the stationary predator. The predators seem to evolve only two kinds of actions, one to go left or right and the other to go up or down. Since the simulated world is toroidal, these two actions are sufficient to catch the prey wherever it may be. Two actions are easier to evolve than five, especially when it also involves evolution of an interpreting system to decipher incoming messages. From the figure 3, it can further be seen that there are straight-line borders between the regions where an upward or downward action is taken and the regions where rightward or leftward actions are taken. This shows that the stationary predator has evolved to send particular messages based on the region in which the blind predator is. It is easier to evolve to recognize a large region with straight-line borders than to identify each grid cell with a different message.

In the second experiment, it can be seen from figure 5 that the predator team has been successful in catching prey. As claimed in the previous section, this indicates that a common

messaging code has been evolved. This is confirmed by analyzing the figures 6: 6a is an exact copy of 6c, and 6b is exactly like 6d. That is, both blind predators react in the same way to any given message. For a given location of a blind predator, channel 1 has evolved to send a particular message. This message is interpreted in the same way by both blind predators. This is true in spite of the fact that the two blind predators, upon close examination, have completely different connection weights in their neural networks.

Figure 6a and 6b do not look alike. This is because the two communication channels may have two different roles to play in the prey capture task. For example, one of them may guide its receiving predator to attack the prey from one direction, while the other channel may direct its predator to go after the prey from the other direction. Thus for any given location of a blind predator, the two communication channels of the stationary predator may transmit different messages. But, as discussed above, the two different messages will be from the same messaging code. Such experiments can also provide clues on the evolution of communication in nature - where perhaps organisms with common goal and similar sensory information evolved a common language.

Another important observation is that the agent controllers are evolved from separate sub-population (unlike some of earlier research). This makes the problem of evolution of consistent language more difficult, since the agents do not share any genes.

Messaging performs slightly better than direct communication as shown in experiment 3. In direct communication, the predators can track each other's position accurately and

thus have more information about the environment (Markovian). However, the predators evolve equally good cooperative behaviors through messaging. The predators can convey information (gathered by sensing prey position) about the environment to each other and thus help disambiguate the non-Markovian environment.

The evolution of language in the form of messages can be put to further use in more complex tasks where just sight (direct communication) will not be sufficient to successfully complete the task. The evolution of prey has also been shown in previous work to lead to more interesting and sophisticated behaviors on the part of both predators and prey (Rawal et al., 2010). Scaling messaging to more agents reliably is another challenge for the future.

Conclusion

The evolution of a messaging code for useful communication was successfully evolved in a team of predators using neuroevolution. If there are two predators communicating with each other, they may develop different codes for sending and receiving. To encourage them to evolve a common communication code, three predators were put on the grid world and their channels of communication with one another were frequently switched. This led to the successful evolution of a single messaging code. This approach to evolution of language can be adopted in more complex and open-ended domains, such as the evolution of realistic and interesting video game agents, or robots.

Acknowledgements

This research has been sponsored by following NSF Grants: DBI-0939454, IIS-0915038.

References

- Agrawal, R. B. and Deb, K. (1994). Simulated binary crossover for continuous search space. Technical report.
- Bullinger, A., Wyman, E., Melis, A., and Tomasello, M. (2011). Coordination of chimpanzees in a stag hunt game. *International Journal of Primatology*, 32:1296–1310. 10.1007/s10764-011-9546-3.
- Holekamp, K. E., Sakai, S. T., and Lundrigan, B. L. (2007). Social intelligence in the spotted hyena (*crocuta crocuta*). *Phil Trans R Soc B.*, 362:523538.
- Jim, K. and Giles, C. L. (2001). How communication can improve the performance of multi-agent systems. In *In Proceedings of Autonomous agents'01*, pages 584–591.
- Knoester, D. B., McKinley, P. K., Beckmann, B., and Ofria, C. (2007). Directed evolution of communication and cooperation in digital organisms. In *Proceedings of the 9th European conference on Advances in artificial life, ECAL'07*, pages 384–394, Berlin, Heidelberg. Springer-Verlag.
- Panait, L. and Luke, S. (2005). Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11:387–434. 10.1007/s10458-005-2631-2.
- Rajagopalan, P., Rawal, A., Miikkulainen, R., Wiseman, M., and Holekamp, K. (2011). The role of reward structure, coordination mechanism and net return in the evolution of cooperation. In *2011 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 258–265.
- Rawal, A., Rajagopalan, P., and Miikkulainen, R. (2010). Constructing competitive and cooperative agent behavior using coevolution. In *Proceedings of the IEEE Conference on Computational Intelligence and Games*, pages 107–114. IEEE Press.
- Saunders, G. M. and Pollack, J. B. (1996). The evolution of communication schemes over continuous channels. In *In*, pages 580–589. MIT Press.
- Tuci, E. and Vicentini, F. (2007). Operational aspects of the evolved signalling behaviour in a group of cooperating and communicating robots.
- Vaughan, T., Ryan, M. J., and Czaplewski, N. (2011). *Mammology*. Jones and Barnett.
- Werner, G. and Dyer, M. (1992). Evolution of communication in artificial organisms. In Langton, C., Taylor, C., Farmer, D., and Rasmussen, S., editors, *Artificial Life II*, pages 659–687, Redwood City, CA. Addison-Wesley Pub.
- Yong, C. H. and Miikkulainen, R. (2009). Coevolution of role-based cooperation in multiagent systems. *IEEE Transactions on Autonomous Mental Development*, 1(3):170–186.

odNEAT: An Algorithm for Distributed Online, Onboard Evolution of Robot Behaviours

Fernando Silva¹, Paulo Urbano¹, Sancho Oliveira² and Anders Lyhne Christensen²

¹LabMAG, Faculty of Sciences, University of Lisbon (FC-UL)

²Institute of Telecommunications, University Institute of Lisbon (ISCTE-IUL)

Lisbon, Portugal

fsilva@di.fc.ul.pt

Abstract

We propose and evaluate a novel approach called Online Distributed NeuroEvolution of Augmenting Topologies (odNEAT). odNEAT is a completely distributed evolutionary algorithm for online learning in groups of embodied agents such as robots. While previous approaches to online distributed evolution of neural controllers have been limited to the optimisation of weights, odNEAT evolves both weights and network topology. We demonstrate odNEAT through a series of simulation-based experiments in which a group of e-puck-like robots must perform an aggregation task. Our results show that robots are capable of evolving effective aggregation strategies and that sustainable behaviours evolve quickly. We show that odNEAT approximates the performance of rtNEAT, a similar but centralised method. We also analyse the contribution of each algorithmic component on the performance through a series of ablation studies.

Introduction

The traditional evolutionary computation algorithm works in a discrete and centralised manner. An external component creates an initial population and is responsible for selecting, mutating and replacing individuals. Evolution is usually performed offline even if the subject of optimisation or design is an embodied agent such as a robot. Traditional evolutionary approaches have a number of shortcomings when evolving robotic controllers. Since evolution is typically conducted offline, controllers need to be transferred to robots post-evolution. Once deployed, the controllers are thus specialised to a particular task and environmental conditions. They are fixed solutions and exhibit limited capacity to adapt to environments and to tasks not seen during evolution.

In order for an evolutionary algorithm (EA) to give robots the capacity to continuously adapt, it has to be run on the robots themselves and execute as they perform their tasks, i.e., evolution must be conducted *online*. The first attempt at truly autonomous online evolution in multi-robot systems was proposed in (Watson et al., 1999) and denominated embodied evolution (EE). EE addresses long-term self-adaptation but relies on robots meeting and exchanging genetic material. Frequent encounters between robots is

difficult to guarantee, especially in large and open environments. After EE, different approaches on online evolution have been proposed (discussed in the next section). Notwithstanding, in such contributions, neuroevolution is limited to evolving weights in fixed-topology artificial neural networks (ANN).

In this paper, we introduce odNEAT, a novel online and distributed version of NeuroEvolution of Augmenting Topologies (NEAT) (Stanley and Miikkulainen, 2002; Stanley, 2004). NEAT is a state-of-the-art neuroevolution (NE) method that evolves the weights and the topology of an ANN. odNEAT shares some features with rtNEAT, a real-time enhancement of NEAT designed for video games (Stanley, 2005). In rtNEAT, game characters are able to evolve online while they are playing against humans. Both NEAT and rtNEAT operate in a centralised manner. odNEAT, on the other hand, is completely decentralised. In odNEAT, robots adapt autonomously on the basis of local information. The EA is distributed across multiple robots which have to solve the same task, either individually or collectively. We demonstrate odNEAT in a simulated experiment where a group of e-puck-like robots (Mondada et al., 2009) running an EA independently, online and onboard, must perform an aggregation task. To the best of our knowledge, the contribution presented here is novel in two aspects: (1) an online and distributed version of NEAT has not been proposed and studied prior to this work; (2) this is the first demonstration of online and onboard evolution where both the weights and the topology of the ANN controllers are under evolutionary control.

Related Work

In this section, we review the background and related work in the online evolution of ANN robotic controllers, as well as the main characteristics of NEAT and rtNEAT.

Online Evolutionary Robotics

The first attempt at truly autonomous online evolution in multi-robot systems, *embodied evolution*, was presented in (Watson et al., 1999). In this approach, each robot carries

only a single genotype and is controlled by the corresponding phenotype — a fixed-topology neural network. Robots probabilistically broadcast a part of their (mutated) genes at a rate proportional to their fitness (Probabilistic Gene Transfer Algorithm, PGTA). Robots that receive gene transmissions incorporate this genetic material in their genome at a rate inversely proportional to their fitness. This way, selection and variation (reproduction) operators are implemented in a distributed manner through the interactions between robots. A variant of this scheme was implemented in (Wischmann et al., 2007) in a predator-prey scenario. The interplay of evolution and lifelong individual learning was investigated as a mean of providing adaptability to novel environmental conditions. Each robot had a maturation period during which no mating/replacement can take place. This mechanism allowed robots to adapt using individual learning before being subjected to any selective pressure. However, within the authors' experimental framework, the effects of learning were not significant. Considering both approaches mentioned above, the main disadvantage is the fact that the embodied evolution was dependent on the exchange of genetic information among the robots. In large environments, where such encounters may be rare, the evolutionary process is therefore prone to stagnation.

A different approach, *encapsulated evolution*, overcomes stagnation by using a time-sharing mechanism. To that end, alternative controllers are executed sequentially and their fitness is measured. Each robot maintains a population of genotypes stored internally and run self-sufficient (and possibly different) EAs locally. Within this paradigm, the robots individually adapt through evolution without the necessity of interacting with other robots. Such approach has been successfully applied to tasks of an individual nature such as phototaxis or obstacle avoidance (Haasdijk et al., 2010; Bredeche et al., 2009).

The two methodologies, embodied evolution and encapsulated evolution, can be combined, leading to a hybrid system similar to an island model (Tanese, 1989). In such a system, each robot acts like an island with genetic information being exchanged through intra-island variation and inter-island migration. An example of such a method is the one presented in (Elfving et al., 2005). In that study, robots have to gather batteries while maintaining a virtual energy level that reflects their task performance. If a robot's energy level reaches 0, offspring is created by mating the current controller with one of the genomes collected during lifetime. In (Usui and Arita, 2003), six Khepera robots evolved an avoidance behaviour. Each physical robot ran an independent EA for a sub-population of virtual agents, evaluated by time sharing. Migrated genomes, broadcasted by other robots, were re-evaluated by the receiving robot.

One of the limitations of existing approaches to online evolution is that neuroevolution solely adjust the weights of the ANN. Previous experimentation to determine a suitable

network topology is therefore necessary. Choosing an inappropriate topology affects the evolutionary process and, consequently, the potential for adaptation. In odNEAT, on the other hand, the network topology is a product of a continuous evolutionary process.

NeuroEvolution of Augmenting Topologies (NEAT)

The NEAT method, introduced by (Stanley and Miikkulainen, 2002) is one of the most prominent neuroevolution (NE) algorithms. The method is capable of optimising both the topology of the network and its connection weights. NEAT acts with global and centralised information like canonical GAs. It has been successfully applied to highly complex problems, such as the double pole balancing, outperforming several methods that use fixed topologies (Stanley, 2004). The high performance of the algorithm is due to three key features: tracking genes with *historical markers* to allow meaningful crossover between topologies, a *niching scheme*, and evolving topologies incrementally from simple initial structures (*complexification*).

The network connectivity is represented through a flexible genetic encoding. Each genome contains of a list of connection genes, each of these referring the two node genes connected. Furthermore, a connection gene encompasses the weight of the connection, a bit indicating if the connection gene is genetically expressed and a *global innovation number* (IN), unique for each gene in the population. INs represent a chronology of the genes introduced. With this feature, the difficulty of matching different network topologies (an NP-hard problem) is avoided and crossover can be performed without *a priori* topological analysis. During crossover, genes with the same historical markings are aligned, to produce meaningful offspring. In terms of mutations, NEAT allows for common connection weights perturbations and structural changes that may lead to the insertion of: (1) a connection gene between two previously unconnected nodes or, (2) a node gene, splitting an old connection into two new connections and disabling the former. Each new gene inserted receives an innovation number. This way, genomes representing networks of different topologies remain compatible throughout evolution because their origin is known.

The niching scheme is composed of two building block: speciation and fitness sharing. Speciation divides the population into non-overlapping sets of similar individuals based on a topological similarity measure. This mechanism protects new structural innovations by reducing competition between individuals representing differing structures and network complexities. In this way, newer structures have time to mature. If a species does not improve for a certain number of generations, it is removed from the population. Explicit fitness sharing dictates that individuals in the same species share the fitness of their niche. The fitness scores of existing members of a species are first *adjusted*, i.e., divided by the

number of individuals in the species. Species then grow or shrink depending on whether their average adjusted fitness is above or below the population average.

The third reason why NEAT often outperforms other NE approaches is the incremental exploration of the search space. The algorithm starts with a uniform population of simple networks with no hidden nodes as in SAGA (Harvey, 1993). Complexity is introduced incrementally as a result of structural mutations. Since only structural mutations that have proven to be fit survive, the exploration of the search space is conducted in an incremental manner.

With the purpose of evolving increasingly complex ANNs online, rtNEAT was introduced (Stanley, 2005). Essentially, rtNEAT is a centralised real-time version of NEAT. rtNEAT contains some differentiating characteristics. While NEAT replaces the entire population at each generation, in rtNEAT one offspring is produced at regular intervals, every n time steps. The worst individual is removed and replaced with a child of a parent chosen from among the best. Unlike NEAT, rtNEAT attempts to keep the number of species constant by adjusting a threshold C_t , which determines the topological compatibility of an individual with a species. When there are too many species, C_t is increased to make species more inclusive; when there are too few, C_t is decreased to be stricter. rtNEAT has shown to preserve the dynamics of NEAT, namely protection of innovation through speciation and complexification (Stanley, 2004).

odNEAT: An Online and Distributed Evolutionary Algorithm

odNEAT runs on a group of agents whose objective is to evolve and adapt while operating in the environment. Each agent is controlled by an artificial neural network that represents a candidate solution to a given task. The evolutionary process takes place online and is an integral part of the agents' behaviour. The typical evolutionary operators (evaluation, selection and reproduction) are carried out autonomously by the agents in the environment without any need for external intervention.

In odNEAT, agents maintain a virtual energy level reflecting their individual performance in the task. The energy level increases and decreases as a result of the agent's behaviour, similarly to the work presented in (Elfwing et al., 2005). If an agent's energy reaches zero, its active chromosome (the genetic encoding of an ANN) is replaced. One general problem, especially for highly complex tasks, is that online evaluation is inherently noisy. Very dissimilar evaluation conditions may be presented to different chromosomes when they become active. Location of embodied agents and the proximity to other agents are factors that directly influence an agent's performance and behaviour. With the purpose of obtaining a reliable fitness estimate, odNEAT distinguishes between the fitness value of an agent and its current energy level. The fitness value is defined as the average of

the energy level, sampled at regular time intervals.

In odNEAT, each agent maintains a local set of chromosomes in an internal repository. The repository is a genetic pool that stores a limited number of chromosomes and their respective fitnesses. The stored chromosomes are arranged into species based on the niching scheme of NEAT. The set of chromosomes include the agent's current and previous active chromosomes and those received from other agents. Each agent probabilistically broadcasts its active chromosome to agents in its immediate neighbourhood, an *inter-agent reproductive event*, with a probability computed as follows:

$$P(event) = \frac{\bar{F}_k}{\bar{F}_{total}} \quad (1)$$

where \bar{F}_k is the average adjusted fitness of *local* species k to which the chromosome belongs and \bar{F}_{total} is the sum of all *local* species' average adjusted fitnesses. Due to the broadcast of genetic information, the active chromosome of an agent may be present in another agent's repository. Such migrations approximate in a distributed manner and over time the reproduction dynamics of rtNEAT. This way, each repository is a local mirror of what happens in the population at large, but no agent has a complete global view of the system.

Besides the internal repository, each agent also maintains a local tabu list, a short-term memory which keeps track of recent *poor* solutions: chromosomes removed from the repository or that caused the robot to run out of energy. Newly received chromosomes must first be accepted by tabu list. The *acceptance condition* is only met if the received chromosomes are topologically dissimilar from all chromosomes in the tabu list.

After the pre-evaluation by the tabu list and if the acceptance condition was met, a received chromosome becomes part of the repository if it has a fitness score higher than the worst local chromosome thus enabling a progressive improvement. Due to the fixed size of the repository, whenever it is full, the insertion of a new chromosome is accompanied by the pre-requisite of removing the chromosome with the *worst adjusted fitness*. When a new chromosome is removed or added, the corresponding species has one less or one more element and therefore the adjusted fitness \bar{F} is recalculated. Whenever an agent receives a copy C' of a chromosome C already contained in the repository (structurally the repository does not allow copies of the same chromosome), the energy level of C' is used to incrementally average the fitness of the C and provide a more reliable indicator of its value.

A particular characteristic of NEAT is the chronology of the genes due global to innovation numbers, which are assigned sequentially. In order to allow a decentralised implementation, odNEAT uses local high-resolution timestamps instead of innovation numbers. Each agent is responsible for

assigning a timestamp to each local innovation, be it a connection or a node. Using high-resolution timestamps for labels practically guarantees uniqueness and allows odNEAT to retain NEAT's concept of chronology.

When an agent's energy reaches zero (because it is incapable of accomplishing the task), a new individual is created. In this process - *an intra-agent reproductive event* - a parent species is chosen with probability proportional to its average fitness, as defined in Equation 1. Then, two parents are selected from the species, each one via a tournament selection of size 2. Offspring is then created based on NEAT's genetic operators: crossover of the parents' genomes and mutation of the new chromosome.

One important aspect regarding newly created individuals is the importance of letting them act in the environment for a minimum amount of time α . This time, denominated as the *maturation period*, gives the new individuals a change to spread their genome by mating with other agents and provides a habituation period. An individual can continue to be active after it reaches α , if its energy is above 0. In Fig. 1, we summarise odNEAT as executed independently by each agent.

```

odNEAT()
  initialize_genes()
  energy = default_energy
  LOOP
    if (broadcast?) then
      send(all_genes, agents_in_range)
    endif
    if (has_received?) then
      for element in received do
        if (tabu_and_repository_accept(element))
          add_to_repository(element)
          adjust_repository_size()
          adjust_species_fitness()
        endif
      endfor
    endif
    act_in_environment()
    energy = update_energy_level()
    if (energy <= 0 AND not (IN_MATURATION_PERIOD?))
      add_to_tabu_list(old_controller)
      generate_offspring()
      assign_as_controller(offspring)
    endif
  ENLOOP

```

Figure 1: Pseudo-code of odNEAT that runs independently on every agent (see text).

Experimental Methodology

To assess odNEAT, we applied the algorithm in a simulated collective robotics experiment. The simulated robots are modelled after the e-puck (Mondada et al., 2009), a small (75 mm in diameter) differential drive robot capable of moving at a maximum speed of 13 cm/s. Each robot is equipped with eight infrared sensors, capable of obstacle detection and communication at a range of up to 25 cm between emit-

ter and receiver.¹ Each infrared sensor is subjected to noise, which is simulated by adding a random Gaussian component within $\pm 5\%$ of the sensor saturation value. Besides these sensors, each robot has an internal energy level sensor and a counter, which allow it to respectively perceive its current virtual energy level and the number of *distinct* chromosomes received during the most recent P control cycles.

The environment consists of a square arena surrounded by walls. The size of the arena was chosen to be 3 x 3 meters. At any time, a robot can thus sense less than 2.90% of the environment. Each of the robots is controlled by an artificial neural network produced by odNEAT. The input layer consists of one neuron for each proximity sensor (detects walls and other robots), one neuron for the energy sensor, and one neuron for the counter. The output layer contains two neurons, one for each wheel of the robot.

Since we are working with a process of continuous evolution, experiments continue until all robots achieve sustainable energy levels or until a temporal upper bound of 100 hours of simulated time is reached, in which case the experiment is considered to have failed. We are primarily interested in: i) determining if odNEAT evolves controllers capable of solving the specified task, ii) the elapsed time, to measure the speed of the evolutionary process and, iii) the quality of the solution and the behaviours evolved, that is, how the robots search through the environment and locate each other.

The Aggregation Task

In an aggregation task, dispersed agents must move close to one another so that they form a cluster. Aggregation plays an important role in many biological systems since it is the basis for the emergence of various collective behaviours. For instance, several social animals use aggregation to increase their chances of survival, or as a pre-cursor of other behaviours. In robotics, self-assembly and collective transport of heavy objects require prior aggregation at the site of interest. Due to the collective nature of the task, the topological (and possible behavioural) heterogeneity of the evolved controllers is an intriguing aspect.

Our experiments were conducted with a group of 5 robots placed in initial random positions at a minimum distance of 1.5 meters between neighbours. At each control cycle, a robot's virtual energy level E is updated according to the following equation:

$$\frac{\Delta E}{\Delta t} = \alpha(t) + \gamma(t) \quad (2)$$

where $\alpha(t)$ is a reward proportional to the number of controllers received in the last time period P (see Table 1). Since

¹The original e-puck infrared range is 2-3 cm (Mondada et al., 2009). In real e-pucks, the *liblrcom* library, available at <http://www.e-puck.org>, allows to extend the range up to 25 cm and multiplex infrared communication with proximity sensing.

information is transmitted locally, this factor indicates the presence of robots nearby. $\gamma(t)$ is a factor related to the quality of movement and rewards robots that are capable of exploring the space in a relatively coordinated manner:

$$\gamma(t) = \begin{cases} -1 & \text{if } v_l(t) \cdot v_r(t) < 0 \\ \Omega_s(t) \cdot \omega_s(t) & \text{otherwise} \end{cases} \quad (3)$$

where $v_l(t)$ and $v_r(t)$ are the left and right wheel speeds and $\Omega_s(t)$ is the ratio between the average and maximum speed achievable. $\omega_s(t) = 1 - \sqrt{|v_l(t) \cdot v_r(t)|}$ rewards robots for setting similar speeds on its two wheels, to avoid any turning-on-the-spot behaviour.

The experimental configuration is presented in Table 1. All parameters were fine-tuned through a trial-and-error process. Regarding NEAT, we have used the default parameters (as specified in Stanley (2005)) except for the crossover and mutation rates. Such parameters take values significantly lower than the default values (25% and 10%, respectively).

Parameter	Values
Repository size	40 chromosomes
Energy (initial/max)	1000/2000 e.u.
$\alpha(t)$	3 e.u. per chromosome
Time period P	10 cycles
Maturation period	500 cycles
Crossover rate	0.25
Mutation rate	0.1
Add node probability	0.03
Add connection probability	0.05
Weight mutation magnitude	0.5
Recurrent connection prob.	0.2
Maximum simulation time	100 hours

Table 1: Configuration for the aggregation experiments. Cycles represent robots' control cycles and e.u. denote energy units. The parameters were fine-tuned through a trial-and-error process.

Results and Discussion

In all 30 evolutionary runs performed, robots managed to evolve behaviours that could effectively explore the environment and keep the energy level above 0. We observed that aggregation into a single group was successfully achieved in 22 of the 30 runs. In the remaining 8 runs, the 5 robots formed two groups, one group of three robots and one group of two robots. In spite of such final configuration, robots still maintained self-sustainable energy levels. They evolve adequate behaviours for searching, locating and joining other robots in the environment. By analysing the details of each experiment, we observed the emergence of two types of strategies: *group clustering* (see Fig. 2) and *individual search* (see Fig. 3) behaviours.

Group Clustering: As a group, robots frequently evolve two distinct strategies (Fig. 2): a *static* and a *dynamic* clustering behaviour. In the static category, robots meet in some part of the environment and, by detecting one another, maintain their relative positions thus leading to a very stable behaviour. The other category, *flocking* or *dynamic* clustering creates loose and moving groups. In this case, robots meet and start moving together to explore the environment. The latter behaviour is less stable than the static clustering. When robots decide to flock and then collide with walls, it provokes a temporary de-synchronization of movement. As a consequence, and due to their short range of sensors, robots may lose sight of one another and will have to restart their search behaviour.

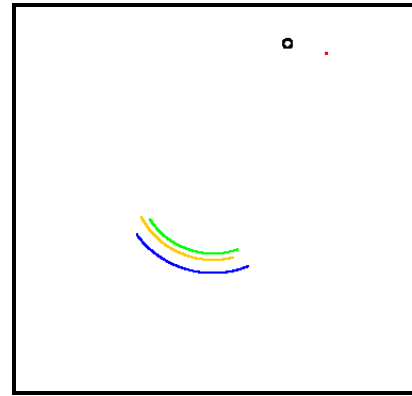


Figure 2: Traces of the robots' group clustering. Three robots exhibit a flocking behaviour while the remaining two form a static cluster, eventually leading to a single aggregate.

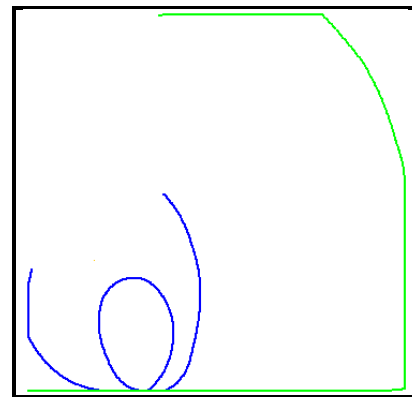


Figure 3: Traces of two of the most frequently evolved individual search strategies. One searches near the walls while the other presents a more circular trajectory thereby covering a larger area.

Individual Strategies: In terms of individual strategies for searching the environment, the evolved behaviours fall in

two categories (Fig. 3). The first one, *navigating near walls*, consists of exploring the environment by moving along the walls of the arena. In some instances of this behaviour, the searching robot moves away from the walls from time to time to explore. The second category consists of behaviours exhibiting a circular trajectory. The searching robot moves across the arena while rotating about itself. This way, the robot is capable of covering a wider area than the walls-based search strategy.

Figure 4 shows the time required to solve the task in each evolutionary run. The highest value is 24.43 hours of simulated time while the lowest is 1.10 hours. On average, each group of 5 robots takes 6.22 ± 5.55 hours of simulated time to aggregate.

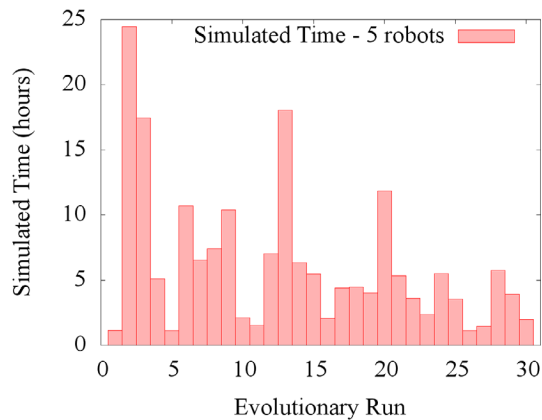


Figure 4: Experimental Results. Simulated time required to accomplish the task in each evolutionary run.

Measure	Average	Minimum	Maximum
Sim. Time	6.22 ± 5.55	1.12	24.45
Evaluations	104 ± 81	25	313

Table 2: Summary of the experimental results with a group of five robots. Time is listed in hours of simulated time.

Considering the average number of evaluations, i.e., controllers per robot, each robot was governed by 104 ± 81 controllers (see Table 2). The variance in the average time and number of evaluations can be explained by the non-linearity of the task. Robots have a short range of sensors and are placed in a large environment. Each robot may be able to search the environment very efficiently but, since it senses less than 2.90% of the total area, the process of finding other robots can be time consuming, especially if we consider that different robots are likely to execute different behaviours for exploring the environment, as happens with the individual search strategies.

rtNEAT and odNEAT

In spite of odNEAT being intentionally distributed, an interesting question is how the results of odNEAT compare to rtNEAT (Stanley, 2005), which relies on traditional centralised evolution. With the purpose of comparing the performance of odNEAT and rtNEAT and thus examine the costs of distributing NEAT, we setup a new series of experiments (with 30 independent runs) with a group of five robots. In order to provide a basis for comparison between the two EAs, two aspects of rtNEAT were altered. First, the dynamic compatibility threshold was fixed, as it is in odNEAT and NEAT. Second, offspring is not created based on a time condition but instead when a robot's energy level reaches zero. In the experiments, rtNEAT operated with a population size of 200 individuals, thus maintaining an average of 40 possible solutions per robot, as in odNEAT.

Method	Sim. Time	Evaluations
odNEAT	6.22 ± 5.55	104 ± 81
rtNEAT	4.44 ± 3.27	96 ± 60

Table 3: Performance comparison between odNEAT and rtNEAT, representing the costs of distributing NEAT. Time is listed in hours of simulated time.

Experimental results are listed in Table 3 and demonstrate the performance costs from distributing NEAT. In comparison with rtNEAT, odNEAT presents a slightly lower performance by requiring each robot to test approximately 8 controllers more (an equivalent to 8.33%). Notice that odNEAT, due to its distributed nature, does not assess the group level information from the global perspective. As a consequence, odNEAT requires more time to evolve solutions. The number of evaluations suggest that odNEAT provides results comparable to rtNEAT. The cost of operating solely based on local information is relatively low. odNEAT has another important advantage over the centralised EAs, namely the lack of dependency on an external mechanism which makes the approach resilient. If a robot fails, for instance, the group can adapt to accommodate for the faulty unit.

Ablation Studies

In order to verify the contribution of each algorithmic component in odNEAT, we performed a series of ablation studies considering the initial group of five robots. In particular, we tested the system's performance in three distinct experimental configurations: (1) without the maturation period, (2) with a minimal internal repository of size 2 and; (3) without the tabu list. Results, present in Table 4, are averaged over 30 independent evolutionary runs for each configuration. Averages in this table exclude runs that failed to find sustainable behaviours within 100 hours of simulated time.

The most critical algorithmic component of odNEAT is the internal repository, an evidence supported by statisti-

Method	Sim. Time	Evaluations	Failure Rate
Min. Repository	16.38 \pm 21.15	236 \pm 214	26.67%
No-tabu	8.88 \pm 12.19	134 \pm 119	6.67%
No-maturation	13.86 \pm 22.53	211 \pm 286	3.33%
Full odNEAT	6.22 \pm 5.55	104 \pm 81	0

Table 4: odNEAT ablations summary. The table lists the average simulation time (in hours), the average number of evaluations and the failure rate of each method, in the aggregation task. Each ablation leads to an inferior and less efficient algorithm.

cal significance ($\rho < 0.003$, Student's t-test). Since we are dealing with a process of continuous evolution, the repository maintains a local view of the system's history and provides the genetic basis for evolution. With a minimal repository, evolution is limited to a small set of chromosomes (in this case 2). In such scenario, the evolutionary process is much slower and may even be incapable of exploring enough of the solution space to find adequate solutions hence the high failure rate, simulation time and number of evaluations. Without the tabu list, odNEAT presents a failure rate of 6.67% but, when evolution is on the right track, it finds solutions relatively fast. Arguably, this means the tabu list keeps the evolutionary process from cycling around in one neighbourhood of the solution space, which sometimes happens due to the fact that robots act based only on local information. The tabu list promotes topological diversity in the repository by rejecting chromosomes similar to those that have already failed. The maturation period defines a lower bound of activity in the environment, giving the individuals a chance to spread their genome. If not for this component, good solutions could potentially be lost forever and evolution would be decelerated. Robots would still be capable of solving the task most of the times. However, they would not be able to and improve their behaviour iteratively through the exchange of genetic information unless they were situated close to each other. Arguably, the most important conclusion that can be drawn from the ablation studies is that all of the parts of odNEAT are necessary to guarantee its performance as an effective online distributed EA.

Scalability Experiments

The impact of the group size on performance was analysed by conducting 30 independent evolutionary runs for groups of 5, 10, 15, 20, 25 and 30 robots. The area of the arena was increased proportionally to the number of robots. Notice that if we maintained the same size of the environment, the experimental setup would not be fair: with the increasing density of robots in the environment, the task would be easier to solve simply because robots would encounter each other more frequently. Table 5 shows the area of the squared arena in each experimental configuration.

Group Size	Arena Area
5	9 m ²
10	18 m ²
15	27 m ²
20	36 m ²
25	45 m ²
30	54 m ²

Table 5: Environment size for each experimental configuration.

Experimental results are listed in Table 6. The time required to accomplish the task increases approximately 36% when the group size was increased from 5 to 10 robots. However, the average number of evaluations, a natural measure of performance, is almost similar except for a higher standard deviation. In fact, the increase in the time required is mainly due to 4 runs, displayed in Table 7. In these cases, robots managed to solve the task mainly by forming small aggregates, of two or three robots. Since small groups are difficult to detect by other robots, robots not belonging to any aggregate required more time to find a group of robots to join and to stabilise their behaviours.

Group Size	Sim. Time	Average Evaluations
5	6.22 \pm 5.55	104 \pm 81
10	8.49 \pm 11.31	112 \pm 117
15	3.71 \pm 3.09	63 \pm 44
20	3.49 \pm 2.79	57 \pm 37
25	3.33 \pm 1.34	55 \pm 22
30	3.78 \pm 2.56	54 \pm 28

Table 6: Summary of the scalability experiments. Time is listed in hours of simulated time.

Run	Sim. Time	Average Evaluations
5	33.87	419
8	27.47	364.6
14	36.68	135
16	43	500

Table 7: Outliers within the 10 robots' evolutionary runs. Time is listed in hours of simulated time.

Further increasing the group size, we observe that the performance improves substantially until it reaches a stable level around a group size of 15 robots. Results show that, for larger groups, the time required to accomplish the task and evolve sustainable behaviours is relatively constant. With the increase in the size of the environment, there is a larger area to search and explore. In relative terms and in spite of the group size increase, the robots sense a smaller portion of the environment. In this scenario, the stable performance

is, in fact, an argument in favour of odNEAT's scalability; the conditions for solving the task become more challenging and the robots are still able to evolve successful behaviours in the same amount of time.

Conclusions and Future Work

In this paper, we have introduced a novel approach called odNEAT, a completely distributed evolutionary algorithm for collective online learning in groups and swarms of embodied agents. We demonstrated odNEAT through a series of simulation-based experiments in which a group of e-puck-like robots evolved aggregation behaviours. Three points are worth mentioning about the experimental results. First, due to the asynchronous and distributed character of odNEAT, robots displayed different strategies for aggregating. Second, the behaviours evolved, static and dynamic clusters, as well as individual search strategies for exploring the environment, were observed simultaneously in the same group of robots. In spite of such behavioural diversity, robots manage to collaborate effectively towards the common goal. Finally, the comparison between rtNEAT and odNEAT suggest that, in spite of being a distributed EA, odNEAT provides results comparable to the standard centralised rtNEAT. The scalability experiments revealed that, for group sizes from 5 to 15 robots, odNEAT scales well considering the time required to achieve sustainable behaviours. For larger groups, odNEAT maintains the performance levels. Ablation studies show that each of the algorithmic components provide a useful contribution to the performance of odNEAT, accelerating evolution and keeping the evolutionary process from cycling around in one neighbourhood of the solution space.

The immediate follow-up work will investigate a broader class of collective tasks in evolutionary robotics. One of the promising directions for odNEAT is to study to what extent agents are capable of continuously adapt in dynamically changing environmental conditions. In the future, we also intend to investigate the basic requirements for truly open-ended evolution, in which the evolutionary process should be capable of producing a large variety of different and novel solutions to a given task.

Acknowledgements

The authors gratefully acknowledge comments and discussions with Kenneth O. Stanley and Luis Correia during the preparation of this paper.

References

- Bredeche, N., Haasdijk, E., and Eiben, A. E. (2009). On-line, on-board evolution of robot controllers. In Collet, P., Monmarche, N., Legrand, P., Schoenauer, M., and Lutton, E., editors, *9th International Conference on Artificial Evolution*, volume 5975 of *Lecture Notes in Computer Science*, pages 110–121. Springer-Verlag, New York, NY.
- Elfwing, S., Uchibe, E., Doya, K., and Christensen, H. I. (2005). Biologically inspired embodied evolution of survival. In *IEEE Congress on Evolutionary Computation*, pages 2210–2216. IEEE Press, Edinburgh, UK.
- Goldberg, D. (2002). *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*, volume 7. Kluwer Academic Publishers, Boston, MA.
- Haasdijk, E., Eiben, A. E., and Karafotias, G. (2010). On-line evolution of robot controllers by an encapsulated evolution strategy. In *Proceedings of the 2010 IEEE Congress on Evolutionary Computation*, pages 1–7. IEEE Press, Piscataway, NJ.
- Harvey, I. (1993). Evolutionary robotics and SAGA: The case for hill crawling and tournament selection. In Langton, C. G., editor, *Proceedings of the Workshop on Artificial Life (ALIFE '92)*, volume 17 of *Sante Fe Institute Studies in the Sciences of Complexity*, pages 299–326. Addison-Wesley, Reading, MA.
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotocz, A., Magnenat, S., Zufferey, J., Floreano, D., and Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, volume 1, pages 59–65. IPCB, Castelo Branco, Portugal.
- Stanley, K. O. (2004). *Efficient Evolution of Neural Networks through Complexification*. PhD thesis, The University of Texas at Austin, Austin, TX.
- Stanley, K. O. (2005). Evolving neural network agents in the NERO video game. In *Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games*, pages 182–189. IEEE Press, Piscataway, NJ.
- Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.
- Tanese, R. (1989). *Distributed Genetic Algorithms for Function Optimization*. PhD thesis, University of Michigan, Ann Arbor, MI.
- Usui, Y. and Arita, T. (2003). Situated and embodied evolution in collective evolutionary robotics. In Sugisaka, M. and Tanaka, H., editors, *Proceedings of the 8th International Symposium on Artificial Life and Robotics*, pages 212–215. Dept. of Electrical and Electronic Engineering, Oita University, Japan.
- Watson, R. A., Ficici, S. G., and Pollack, J. B. (1999). Embodied evolution: Embodying an evolutionary algorithm in a population of robots. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC 99)*, pages 335–342. IEEE Press, Piscataway, NJ.
- Wischmann, S., Stamm, K., and Wörgötter, F. (2007). Embodied evolution and learning: The neglected timing of maturation. In e Costa, F. A., Rocha, L. M., Costa, E., Harvey, I., and Coutinho, A., editors, *European Conference on Artificial Life*, volume 4648 of *Lecture Notes in Computer Science*, pages 284–293. Springer-Verlag, Berlin, Germany.

Finding Optimal Random Boolean Networks for Reservoir Computing

David Snyder¹, Alireza Goudarzi², and Christof Teuscher³

¹Department of Computer Science

²Systems Science Graduate Program

³Department of Electrical and Computer Engineering

Portland State University, Portland, OR 97201

¹dsnyder@cecs.pdx.edu ²alirezag@cecs.pdx.edu ³teuscher@pdx.edu

Abstract

Reservoir Computing (RC) is a computational model in which a trained readout layer interprets the dynamics of a component called a reservoir that is excited by external input stimuli. The reservoir is often constructed using homogeneous neural networks in which a neuron's in-degree distributions as well as its functions are uniform. RC lends itself to computing with physical and biological systems. However, most such systems are not homogeneous. In this paper, we use Random Boolean Networks (RBN) to build the reservoir. We explore the computational capabilities of such a RC device using the temporal parity task and the temporal density classification. We study the sufficient dynamics of RBNs using *kernel quality* and *generalization rank* measures. We verify findings by Lizier et al. (2008) that the critical connectivity of RBNs optimizes the balance between the high memory capacity of RBNs with $\langle K \rangle < 2$ and the higher information processing of RBNs with $\langle K \rangle > 2$. We show that in a RBN-based RC system, the optimal connectivity for the parity task, a processing intensive task, and the density classification task, a memory intensive task, agree with Lizier et al.'s theoretical results. Our findings may contribute to the development of optimal self-assembled nanoelectronic computer architectures and biologically-inspired computing paradigms.

Introduction

In this paper, we propose discrete Boolean networks with heterogeneous in-degrees and transfer functions for Reservoir Computing (RC). Reservoir computing is an emerging paradigm in Recurrent Neural Networks (RNN) (Haykin, 2009) that

promotes computing using intrinsic dynamics of an excited system called the reservoir (Lukosevicius and Jaeger, 2009). The reservoir acts as a temporal kernel function, projecting the input stream into a higher dimensional space, thereby creating features for the readout layer. To produce the desired output, the readout layer performs a dimensionality reduction on the traces of the input signal in the reservoir. A system with sufficiently rich dynamics can remember perturbations by an external input over time. Two advantages of using RC are: computationally inexpensive training and flexibility in reservoir implementation. This makes RC suitable for emerging unconventional computing paradigms, such as computing with physical phenomena (Fernando and Sojakka, 2003) and self-assembled electronic architectures (Teuscher et al., 2009). Maass et al. (2002) initially proposed a version of RC called Liquid State Machine (LSM) as a model of cortical microcircuits. Independently, Jaeger (2001) introduced a variation of RC called Echo State Machine (ESM) as an alternative RNN approach for control tasks. Variations of both LSM and ESM have been proposed for many different machine learning and system control tasks (Lukosevicius and Jaeger (2009)). Büssing et al. (2010) conducted a comprehensive study of reservoir performance using different metrics as a function of connectivity K , the logarithm of the number of states per node m , and the variance of the weights in the reservoir. They concluded that for binary networks the performance of an RC system is max-

imized for sparse reservoir networks, but as the number of states per node increases the performance becomes insensitive to sparsity. Insofar, most of the RC research is focused on reservoirs with homogeneous in-degrees and transfer functions. However, due to high design variation most self-assembled systems are heterogeneous in their connectivity and transfer functions. Wang et al. (2006) introduced a hybrid RC device that uses both sigmoidal and wavelet nodes and showed that it improved the reservoir performance. Here, we use a well-known simple heterogeneous Boolean network model for the reservoir. Kauffman (1969) first introduced this model to study gene regulatory networks. Kauffman showed these Boolean networks to be in a complex dynamical phase at “the edge of chaos” when the average connectivity (in-degree) of the network is $\langle K \rangle = 2$ (critical connectivity). Rohlf et al. (2007) showed that near critical connectivity information propagation in Boolean networks becomes independent of system size. Goudarzi et al. (2012) studied adaptive computation and task solving in Boolean networks and found that learning drives the network to the critical connectivity $\langle K \rangle = 2$. Here, we show that heterogeneous discrete Boolean networks in the super-critical regime ($\langle K \rangle > 2$) can be used as the reservoir to perform non-trivial computation. To the best of our knowledge this is the first time that discrete, heterogeneous dynamical networks have been used in reservoir computing.

Experimental Setup

Model

Structurally, RC is made up of three parts: input layer, reservoir or kernel, and readout layer. The input layer excites the reservoir by passing the input signal to it and the readout layer interprets the traces of the input signal in reservoir dynamics to compute the desired output. In our model, the reservoir is a Random Boolean Network (RBN). The fundamental subunit in a RBN is a node with K input connections. At any instant in time, the node can assume either of the two binary states “0” or “1.” The node updates its state at time t according to a K -to-1 Boolean mapping of its K inputs.

Therefore, the state of a single node at time $t + 1$ is only determined by its K inputs at time t and by one of the 2^{2^K} Boolean functions used by the node. Formally, a RBN is a collection of N such binary nodes. For each node i out of N nodes, the node receives K_i inputs, each of which is connected to one of the N nodes in the network. In this model, self-connections are allowed.

The network is random in two different ways: 1) the source nodes for an input are chosen from the N nodes in the network with uniform probability and 2) the Boolean function of node i is chosen from the $2^{2^{K_i}}$ possibilities with uniform probability. Each node sends the same value on all of its output connections to the destination nodes. The average connectivity will be $\langle K \rangle = \frac{1}{N} \sum_{i=1}^N K_i$. We study the properties of RBNs characterized by N nodes and average connectivity $\langle K \rangle$. This refers to all the instantiations of such RBNs. Once the network is instantiated, the collective time evolution at time t can be described as using $x_i^{t+1} = f_i(x_1^t, x_2^t, \dots, x_{K_i}^t)$, where x_i^t is the state of the node i at time t and f_i is the Boolean function that governs the state update of the node i . The nodes are updated synchronously, i.e., all the nodes update their state according to a single global clock signal.

From a graph theoretical perspective, a RBN is a directed graph with N vertices and $L = \lfloor \langle K \rangle N \rfloor$ directed edges. We construct the graph according to the random graph model (Erdős and Rényi, 1959). We call this model a heterogeneous RBN because each node has a different in-degree. In the classical RBN model, all the nodes have identical in-degrees and therefore are homogeneous. The original model of Kauffman (1969) assumes a static environment and therefore does not include exogenous inputs to the network. To use RBNs as the reservoir, we introduced I additional input nodes that distribute the input signals to the nodes in the network. The source node of K_i links for each node i is randomly picked from $N + I$ nodes with uniform probability. The input nodes are not counted in calculating $\langle K \rangle$. For online computation, the reservoir is extended by a separate readout layer with O nodes. Each node in the readout

layer is connected to each node in the reservoir. The output of the node o in the readout layer at time t is denoted by y_o^t and is computed according to $y_o^t = \text{sign} \left(\sum_{j=1}^N \alpha_j x_j^t + b \right)$. Parameters α_j are the weights on the inputs from node j in the reservoir to node o in the readout layer and b is the common bias for all the readout nodes. Parameters α_j and b can be trained using any regression algorithm to compute a target output (Jaeger, 2001).

Measures

Kernel Measurements We characterize the quality of the reservoir by measuring *Kernel Quality* (KQ) and *Generalization Rank* (GR). KQ measures how well the reservoir separates different input streams and GR measures how well the reservoir classifies similar input streams in the same class. This is called the *separability* property (Maass et al., 2002). Büsing et al. (2010) introduced kernel quality KQ as a practical measure that can directly quantify this requirement in any given reservoir. Kernel quality is formally defined as the rank ρ of the matrix \mathcal{M} of which columns are reservoir states after being driven by random input signals. To measure this quantity for a reservoir with N nodes, we first create S random input signals of length \mathcal{T} , where $S = N$ so that a square matrix is formed. We drive the reservoir by each input signal for \mathcal{T} time steps and store the state of each node in the reservoir in a column of \mathcal{M} and calculate $\rho(\mathcal{M})$. GR is calculated the same way as KQ , but using input streams in which the last τ bits are identical. Thus reservoirs with optimal separability will have a high KQ and a low GR . We identify the optimal separability by finding the class of reservoirs in which the difference $\Delta = |KQ - GR|$ is maximal.

Tasks We used the temporal parity and density classification tasks to test the experimental performance of the reservoir systems. According to the task, the RC system is expected to continuously evaluate n bits which were injected into the reservoir beginning at $\tau + n$ time steps in the past. Each task necessitates knowledge of the entire window in addition to its unique requirements. The parity

task demands more processing, while in the relatively simple density task, memory of the input is more significant.

Temporal Parity The task determines if n bits $\tau + n$ to τ time steps in the past have an odd number of “1” values. Given an input stream u , where $|u| = \mathcal{T}$, a delay τ , and a window $n \geq 1$,

$$PAR_n(t) = \begin{cases} u(t - \tau) & : n = 1 \\ \oplus_{i=0}^{n-1} u(t - \tau - i) & : \text{otherwise} \end{cases}$$

where $\tau + n \leq t \leq \mathcal{T} - \tau - n$.

Temporal Density The task determines whether or not an odd number of bits $\tau + n$ to τ time steps in the past have more “1” values than “0.” Given an input stream u , where $|u| = \mathcal{T}$, a delay τ , and a window $n = 2k + 1$ where $k \geq 1$,

$$DNS_n(t) = \begin{cases} 1 & : 2 \sum_{i=0}^{n-1} u(t - \tau - i) > n \\ 0 & : \text{otherwise} \end{cases}$$

where $\tau + n \leq t \leq \mathcal{T} - \tau - n$.

Training and Evaluation For every system, we generate 150 random input streams of $\mathcal{T} = 10$ to comprise a training set, and likewise, 150 randomly generated input streams for the testing set. We train the output node with a form of gradient descent in which the weights of the incoming connections are adjusted after every time step in each training example. Given our system and tasks, this form of gradient descent appears to yield better training and testing accuracies than the conventional forms. We use a learning rate $\eta = 0.01$, and train the weights for up to 20,000 epochs. The accuracy of a single training or testing stream is determined by the number of times that the output of the network at time t matches the expected output as specified by the task divided by the total number of values in the output stream. The accuracy on each input set is summed together, and divided

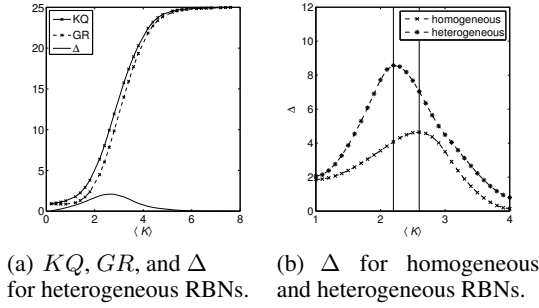


Figure 1: (a) shows the kernel quality KQ , the generalization rank GR , and their difference $\Delta = |KQ - GR|$ of RBN reservoirs with $N = 25$ and $0.10 \leq \langle K \rangle \leq 7.90$. We find the optimal connectivity $\langle K \rangle = 2.25$ at which Δ is maximum. (b) shows the difference Δ for RBNs with homogeneous ($- \times -$) and heterogeneous ($- * -$) in-degree distribution. The peak of the difference Δ predicts greater computational power for heterogeneous RBNs at all connectivities, and is maximum at a lower average connectivity $\langle K \rangle = 2.25$.

by the total number of input streams in the set to produce either the training accuracy T or testing accuracy (generalization capability) G . We are interested in finding the optimal average connectivity $\langle K \rangle$ that maximizes G .

Results

Reservoir Quality

We conduct a comprehensive study of kernel quality and generalization rank in RBN reservoirs for different system sizes N and average connectivity $\langle K \rangle$. Despite the fundamental difference between the reservoirs used in our study and the ones used in (Büsing et al., 2010) (see Models), we find similar behavior in kernel quality KQ and generalization rank GR of RBN. Figure 1 shows KQ and GR for RBNs with size $N = 25$ as a function of the average connectivity $\langle K \rangle$. We used a Spline fit through the data points to create an approximate model for KQ , GR , and Δ .

While both KQ and GR follow a transition-like curve, where near $\langle K \rangle \approx 2.0$ they undergo a sud-

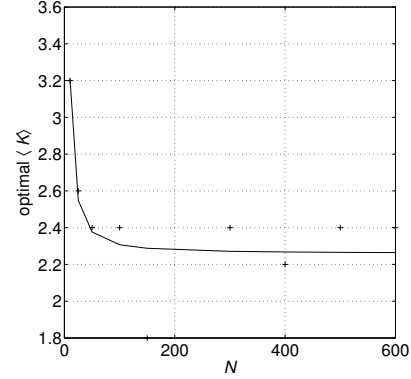


Figure 2: The optimal connectivity for finite system sizes $10 \leq N \leq 600$. The best fit through the data point shows a power-law decay of optimal connectivity with increasing system size (N).

den jump, Δ only shows a maxima at $\langle K \rangle = 2.25$ and decreases for $\langle K \rangle \geq 2.25$. This peak in Δ value represents an optimal connectivity for RBN reservoirs that optimizes both memory and information processing (Lizier et al., 2008).

The Δ on Figure 1(a) corresponds to RBNs with binomial in-degree distribution (heterogeneous). To understand the effect of in-degree distribution we compare the Δ calculation for RBNs with uniform in-degree distribution (homogeneous). Figure 1(b) shows the curve of Δ for different $\langle K \rangle$. We see that for heterogeneous RBNs, Δ peaks at $\langle K \rangle = 2.25$, whereas for homogeneous RBNs, Δ peaks at $\langle K \rangle = 2.65$. Also, the value of the Δ for heterogeneous RBNs is twice as high as the Δ for homogeneous systems, which signifies a greater separation ability of the RBN reservoir (Büsing et al., 2010).

The significance of the optimal connectivity that maximizes the difference Δ in RC is twofold: increased memory capacity and high classification power. Suitable reservoirs need to retain the perturbations from the past input signals to be able to make computations in time. In addition to memory capacity, the optimal connectivity increases the ability of the reservoir to differentiate the input signals that are very different to each other while

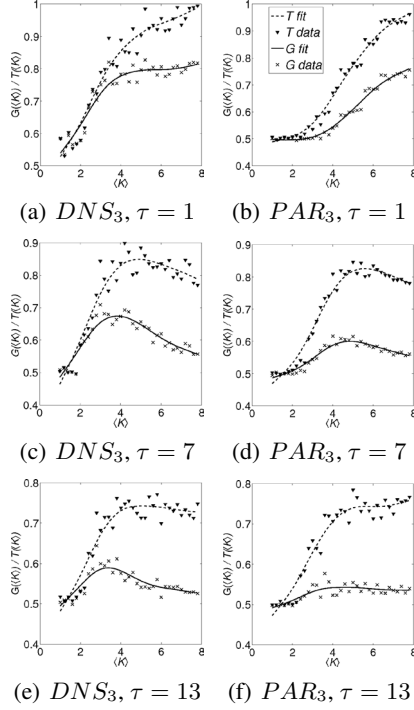


Figure 3: Accuracies for a window size of $n = 3$ and a system size of $N = 500$ as a function of τ and $\langle K \rangle$. Dashed lines illustrate the training accuracy whereas solid lines correspond to the generalization accuracy. The best generalization decreases as τ increases, but favors increasingly small $\langle K \rangle$.

classifying similar (but non-identical) inputs to the same class.

Task Solving

To establish the usefulness of the reservoir quality measurements in RBN reservoirs and to determine the computational power of these RC systems, we use the temporal parity PAR_n and density classification DNS_n tasks. We create reservoirs of various size N with connectivity $\langle K \rangle$ to solve the temporal tasks with different time delays, τ . The temporal parity and the temporal density tasks are both complex tasks that require a knowledge of the entire window of bits over which the

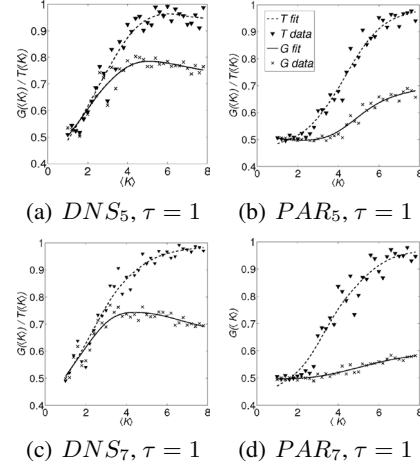


Figure 4: Generalization accuracy for DNS_n and PAR_n tasks for $n = 5$ and $n = 7$. For DNS_n with high window size n , the optimal $\langle K \rangle$ decreases as n increases. This behavior is analogous to what we observe for tasks requiring smaller n and larger τ . For PAR_n the difficulty increases for high n and thus the optimal connectivity increases. Dashed lines illustrate the training accuracy T whereas solid lines correspond to the generalization accuracy G .

task is being computed. However, solving the parity task requires the reservoir to remember all the incoming bits perfectly, otherwise the performance cannot be better than chance. On the other hand, the density task is more decomposable and a classifier with imperfect knowledge of the input bits may still predict a correct answer if it learns the underlying structure of the task. This decomposability of the tasks can be easily characterized using information theoretical reconstructability analysis (Zwick, 2004) of the truth table of the tasks (Goudarzi et al., 2012). Figure 3 shows how the interplay between task complexity, memory capacity and classification capability of the reservoir affects the training and generalization performance in RBN-based RC. We found that larger system sizes accentuate the generalization accuracy trends and so here we use a reservoir of size $N = 500$.

For both tasks we see that the optimal training

and generalizations for small time delay τ occur at high average connectivity $\langle K \rangle = 8$. As the time delay τ increases, the optimal generalization and training accuracies are at lower values of $\langle K \rangle$. We observe that the shift in the optimal connectivity towards lower $\langle K \rangle$ is faster for the density task than the parity task as τ increases from 1 to 13. We also observe that higher connectivity $\langle K \rangle$ causes the training process to overfit the reservoir dynamics. This becomes obvious by reduced generalization accuracy while the training accuracy stays constant or increases. We combined the generalization trends seen in Figure 3 as well as those of other values of τ to construct generalization surfaces for density and parity as functions of $\langle K \rangle$ and τ , using polynomial interpolation of the data points. We observe that the generalization accuracy of the parity task is very sensitive to both average connectivity $\langle K \rangle$ and time delay τ while the generalization accuracy of the density task is more robust to the decrease in $\langle K \rangle$.

The effect of increasing the window size n is similar to increasing τ . In Figure 4(a) and Figure 4(c) we see that the generalization ability of the networks computing DNS_5 and DNS_7 is greatest when $\langle K \rangle$ is lower than in the optimal connectivity shown for DNS_3 in Figure 3(a). We find that the optimal properties of the reservoir on the task DNS_3 when subject to $\tau = 5$ are similar to those of DNS_7 with $\tau = 1$. Increasing τ and n both put more demand on the memory of the reservoir. This drives the optimal connectivity towards low $\langle K \rangle$, where persistent memory is higher. On the other hand, for PAR_n the difficulty of the computation increases exponentially and pushes the optimal connectivity to higher $\langle K \rangle$.

We can explain the observable trends of generalization accuracy as a function of $\langle K \rangle$ and τ using the task complexity and kernel quality measurements. In RC, the reservoir is a collection of coupled filters. Successful computation in RC depends on the reservoir's ability to transform the fluctuations in the input signal into correlated fluctuations in the reservoir. If enough nodes in the reservoir can be perturbed by the input signal, the readout layer will be able to find a suitable map-

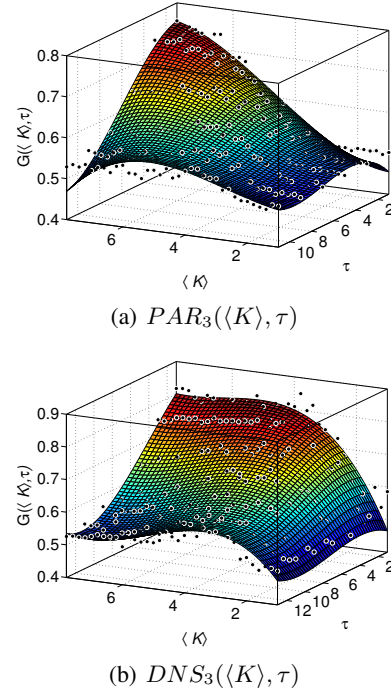


Figure 5: Memory of prior input fades over time, so the generalization accuracy of the RC system diminishes as τ increases. Tasks which require a reservoir to remember old inputs perform best at a $\langle K \rangle$ which is smaller than the optimal $\langle K \rangle$ for those reservoirs that have less time to process input.

ping between the reservoir state and the target output. If the desired task is highly nonlinear, the readout layer will require more correlated variations in the reservoir to be able to find the right mapping. In addition to task complexity, the low time delay requirement favors reservoirs with high connectivity. Solving a task with low time delay implies that the fluctuations in the input signal should be propagated to enough nodes in the reservoir within the required time delay τ , which is true in networks with higher connectivity. On the other hand, tasks with a long time delay need to extract and manifest the fluctuations in the input signal after longer time intervals and therefore require less connectivity. For long-time-delay tasks,

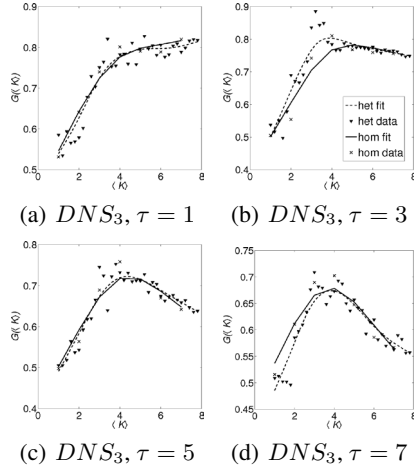


Figure 6: Generalization accuracies $G(\langle K \rangle)$ for DNS_3 of homogeneous in-degree RBNs superimposed over the corresponding measurement for heterogeneous RBNs.

higher connectivity in networks results in fast percolation of the input signal followed by rapid distortion of the correlations between reservoir dynamics and the past signal due to the new input. In other words, memory of the older inputs will be quickly wiped by the new inputs. Therefore, high connectivity networks achieve higher information processing while having lower memory capacity and low connectivity networks have lower information processing power and higher memory capacity (Lizier et al., 2008). This trade-off between information processing and memory explains the generalization accuracy trends.

For the more complex task of parity with a low time delay, only the reservoirs with high connectivity may extract the required features of the inputs instantaneously (i.e., $\tau = 1$). As we increase τ , the reservoir requires a lower connectivity to solve the task. We observe the same trend in the density task as well.

We test the generalization accuracy $G(\langle K \rangle)$ of the homogeneous RBN reservoir for the DNS_3 task and we find that the generalization performance of the RBN reservoir is invariant against the in-degree heterogeneity (see Figure 6).

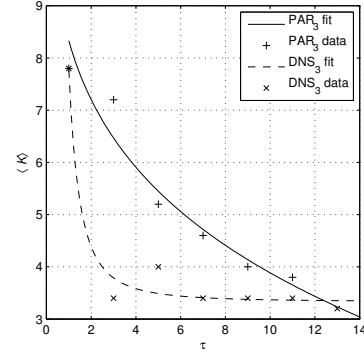


Figure 7: Optimal connectivity of RBN reservoirs for online computation as a function of time delay τ . Solid and dashed lines correspond to the optimal connectivity for the temporal parity and the temporal density tasks respectively.

Discussion

Our results show complex interactions in real-time computation in RC between task complexity, time delay τ , and average connectivity $\langle K \rangle$. We see that optimal connectivity for online computation is a function of the task complexity and the time delay required by the task. We study the change in optimal connectivity as a function of τ by plotting the connectivity of the highest G for different τ (Figure 7). The solid and dashed lines of Figure 7 correspond to the parity and the density tasks respectively. Both curves show that lower connectivity reservoirs are more suitable for online computation as the time delay increases. This lower $\langle K \rangle$ which produces the optimal G in memory intensive tasks may approach that of the $\langle K \rangle$ found to achieve the optimal separability (see Reservoir Quality). The optimal connectivity for the density task follows a sharper decrease for higher τ , due to its relative simplicity in comparison to the parity task (see Task Solving). The parity task is highly nonlinear. Even for high τ , which demands greater memory, the reservoir requires higher connectivity to be able to extract features in a way that are classifiable by the linear readout layer. As in the density task, the parity task requires lower connec-

tivity when τ is high. However, for $1 < \tau < 13$, the optimal $\langle K \rangle$ of PAR_n is higher than that of DNS_n , for all $n > 1$ that we explored.

Conclusion

In this study, we investigated RC using RBNs and observed that for online computations, RBNs show a trade-off between memory capacity and information processing for different average connectivity $\langle K \rangle$. Because of the flexibility of reservoir implementations, RC lends itself to computing with unconventional devices. RC has been proposed for computing with preexisting systems, and while many of these systems have high variation in their individual components, investigation has been primarily towards reservoirs of homogenous networks, both in functions as well as connectivity. By investigating the optimal connectivities of discrete, heterogeneous networks, we have begun to bridge the gap between those reservoirs commonly investigated, and those that could model natural phenomena. In a future study, we will investigate the relation between optimal connectivity and task complexity. The information-theoretical framework developed in (Prokopenko et al., 2011) may help us explore the connection between optimal computation and the nature of the dynamical phase transition in the reservoir.

Acknowledgements

The work was supported by PSU's MCECS Undergraduate Research & Mentoring Program as well as NSF grants #1028120 and #1028378.

References

- Büsing, L., Schrauwen, B., and Legenstein, R. (2010). Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. *Neural Computation*, 22(5):1272–1311.
- Erdős, P. and Rényi, A. (1959). On random graphs. *Publ. Math. Debrecen*, 6:290–297.
- Fernando, C. and Sojakka, S. (2003). Pattern recognition in a bucket. In *Proceedings of the 7th European Conference on Advances in Artificial Life (ECAL 2003)*, volume 2801 of LNCS, pages 588–597. Springer.
- Goudarzi, A., Teuscher, C., Gulbahce, N., and Rohlf, T. (2012). Emergent criticality through adaptive information processing in boolean networks. *Phys. Rev. Lett.*, 108:128702.
- Haykin, S. (2009). *Neural Networks and Learning Machines (3rd Edition)*. Pearson Education, Inc., New York, NY.
- Jaeger, H. (2001). The “echo state” approach to analysing and training recurrent neural networks. Technical Report GMD Rep. 148, St. Augustin: German National Research Center for Information Technology.
- Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467.
- Lizier, J., Prokopenko, M., and Zomaya, A. (2008). The information dynamics of phase transitions in random Boolean networks. In *Eleventh International Conference on the Simulation and Synthesis of Living Systems (ALife XI)*, pages 374–381, Cambridge, MA, USA. MIT Press.
- Lukosevicius, M. and Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149.
- Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–60.
- Prokopenko, M., Lizier, J. T., Obst, O., and Wang, X. R. (2011). Relating fisher information to order parameters. *Phys. Rev. E*, 84:041116.
- Rohlf, T., Gulbahce, N., and Teuscher, C. (2007). Damage spreading and criticality in finite random dynamical networks. *Phys. Rev. Lett.*, 99(24):248701.
- Teuscher, C., Gulbahce, N., and Rohlf, T. (2009). An assessment of random dynamical network automata. *International Journal of Nanotechnology and Molecular Computation*, 1(4):58–73.
- Wang, S., Yang, X.-J., and Wei, C.-J. (2006). Harnessing non-linearity by sigmoid-wavelet hybrid echo state networks (SWHESN). In *The 6th World Congress on Intelligent Control and Automation (WCICA 2006)*, volume 1, pages 3014–3018.
- Zwack, M. (2004). An overview of reconstructability analysis. *Kybernetes*, 33(5/6):877–905.

Representational Momentum May Explain Aspects of Vowel Shifts

Samarth Swarup¹ and Corrine McCarthy²

¹Virginia Bioinformatics Institute, Virginia Tech, Blacksburg, VA 24061

²Linguistics Program / Department of English, George Mason University, Fairfax, VA 22030
swarup@vbi.vt.edu, cmccart6@gmu.edu

Abstract

We present a computational model of vowel shifts, applied in particular to the Northern Cities Vowel Shift. Our model incorporates several empirically-derived rules of vowel change. The key aspect of this model is the use of representational momentum, which, we argue, explains multiple observed features of the shift. We compare our model with data on the Northern Cities Shift spanning more than a century and show that, when representational momentum is included, the results of the model match the data well.

Introduction

Language is a fundamentally social complex system, exhibiting organization at multiple spatial, temporal, and structural levels (Beckner et al., 2009).

Language changes at each of these levels, at various time-scales. For example, syntactic changes tend to be very slow, with significant changes taking centuries, while lexical changes can be very fast, often sweeping across social groups in months or years. This is perhaps intuitive, because a change to the syntax signals a change to the entire language, such as the transition from old to modern English.

Some changes, however, take place on an intermediate timescale of ~100-150 years. This is particularly true of phonological changes, such as vowel shifts.

From the point of view of collective dynamics, vowel shifts are interesting for multiple reasons. They are thought to be initiated when two populations come into contact that have vowel systems shifted with respect to each other. This sets off a complex series of changes as people (mostly subconsciously) adjust their vowel systems in social interactions. Thus, a vowel shift can be seen as convergence on a shared vowel system through a social contagion process.

However, the time-scale of the shift, being over a century, is longer than the lifespan of most humans; this entails that multiple generations of speakers participate in this sound change. Additionally, for the most part, people only adjust their vowel system at a young age, typically until reaching adulthood. This raises the question, what keeps the vowel shift going from one generation to the next, and why do

these shifts appear to stop once they have reached a certain point?

In the present work, we develop an agent-based model of the mechanism of vowel change based on empirical observations by sociolinguists over many years. These observations are incorporated into our model as a system of constraints and update rules that try to preserve these constraints. One key addition we make to the model is to include representational momentum, which is a psychologically documented phenomenon wherein people's memories of sequences of tones ordered by pitch exhibit overshoot. We demonstrate that if we include momentum in the mechanism of individual vowel change, it can account for the long-term population-wide shift in vowels.

The mechanism of vowel change in our model is a phenomenon known as *accommodation*, which is an attempt on the part of a hearer to adjust his vowel system to match the speaker's.

We apply our model to data on the Northern Cities Vowel Shift (NCVS), and show that when representational momentum is included, the results of the model match the data well, while they do not when representational momentum is not included.

The rest of this article is organized as follows. We first present a brief overview of the Northern Cities Shift and the data. Then we discuss prior efforts to model vowel shifts. After that we present our model and experiments with and without representational momentum. We do a statistical analysis of our results to show how they compare with the data, and we end with a discussion of aspects of the shift that are and aren't explained by our model.

The Northern Cities Vowel Shift

The NCVS is a sound change that is currently affecting the vowels of speakers in the cities along the Great Lakes region of the United States (Labov et al., 2006, henceforth LAB). Its evolution over the last 100 years in Chicago, the largest of the Northern Cities, has recently been documented by McCarthy (2009, 2011).

The NCVS is an example of a chain shift: a series of in-

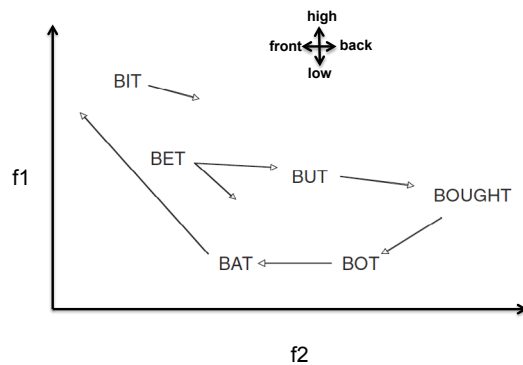


Figure 1: A schematic of the Northern Cities Vowel Shift.

tegrated movements of two or more phonemes. In a chain shift, sounds change in their place of articulation in order to avoid merger and maintain contrast (Labov, 1994; Martinet, 1955).

Consider the example in eq. (1), a minimal chain shift involving two elements that shift in response to one another:

$$/A/ \rightarrow /B/ \rightarrow \quad (1)$$

If the entering element, /A/, shifts first, and /B/ shifts to avoid merger, it is called a push chain. If the exiting element, /B/, shifts first, and /A/ shifts to occupy vacant space, it is called a pull chain (Labov, 1994, 118-119).

The shifting elements of the NCVS are illustrated in figure 1. The axes plot the first two formants (peaks in the frequency spectrum) of the vowels, with the first formant (f1, referred to as “vowel height”) being on the y-axis and the second formant (f2, referred to as “vowel backness”) on the x-axis. Plotted this way, the diagram corresponds roughly to the position in the mouth where each vowel is articulated. Using the first two formants to describe the vowels is standard practice in phonetics (Labov, 1994). Each word corresponds to a vowel class, e.g. BAT represents the vowels of *bad*, *trap*, *cash*, *happy*, etc. Note that front is to the left and back is to the right. The stages of the NCVS, as proposed by LAB and Labov (1994), are as follows:

1. BAT was the first to shift. It went from its traditional, low-front position (IPA [æ]) to one where the nucleus is raised to mid-high position, and followed by an inglide. Pronunciation varies among [eə] ~ [ɪə] ~ [iə].
2. BOT moves forward from the low-back position ([ɑ]) toward the position vacated by low-front BAT.
3. BOUGHT lowers and moves to the front to occupy the space vacated by BOT.
4. BET moves backward toward [ʌ], which is occupied by BUT. BET may also lower toward low-central position.
5. BUT moves backward to avoid collision with BET.
6. BIT lowers to fill the space vacated by BET.

Chronology of the NCVS

Not all authors concur with Labov’s (1994) and LAB’s ordering of events in the NCVS. Citing data from recordings of speakers born in the 1890s, McCarthy (2009) argues that fronted BOT (“front” as in fig. 1) is robust, but raised BAT is absent from the oldest speakers (see also Thomas, 2001). Evidence for BAT raising first appears in those speakers born in the 1910s. BAT shows rapid shifting from a low position to high position in a span of only about 20 years. McCarthy’s and Thomas’s chronology would mean that the NCVS started as a push chain, with BOT pushing BAT. We will address this issue further in our experiments.

The apparent-time construct

LAB’s chronology is based on interviews with speakers of different ages. LAB rely on the apparent-time construct: the assumption that language stabilizes after early adulthood, which means that by sampling speakers of different ages, it is possible to observe language change. Sampling the speech of a 60-year-old is the same as sampling a 20-year-old 40 years ago. Sampling the speech of a 40-year-old is the same as sampling a 20-year-old 20 years ago.

In contrast, McCarthy (2009) and Thomas (2001) incorporate real-time data into their analyses, by analyzing recordings made in the 1960s-1980s of speakers who were born prior to 1900.

Linguistic data for the present study

The present study incorporates both real- and apparent-time linguistic data in order to document a span of 100 years of sound change in Chicago. All speakers were born and raised in the Chicago area, and had parents also from the area.

The real-time data come from archived recordings interviews with six Chicagoans born between 1890 and 1919. Four were gathered as part of the Dictionary of American Regional English project, and two were drawn from Studs Terkel’s interviews, which were digitized by the Chicago History Museum.

The apparent-time data come from sociolinguistic interviews with 35 Chicago-area residents. These speakers read from a list of words containing all the elements of the NCVS.

Speakers were divided into 5 groups based on their birth year. Number of speakers for each age group is as follows: 1890-1910: 3, 1911-1930: 4, 1931-1950: 8, 1951-1970: 10, and 1971-1990: 16.

Vowels were analyzed for f1 and f2 at the point of inflection or midpoint of the steady state, as described in LAB. f1/f2 measurements were normalized using Labov’s G method (Thomas and Kendall, 2007) in order to control for differences in vocal tract size between men and women as much as possible. The data are plotted in fig. 2.

Despite the sample sizes being quite small for the oldest groups, we can make some general observations from the data. First, though Labov claims that BAT raised before

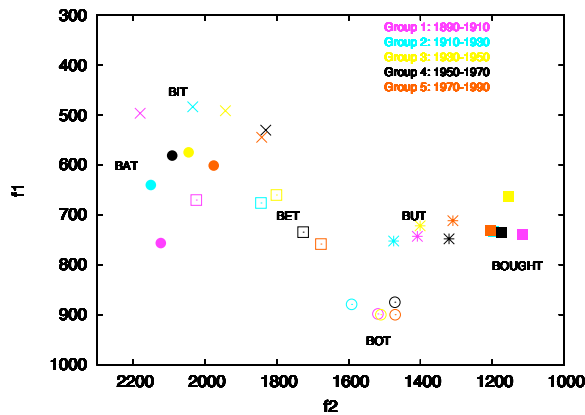


Figure 2: Mean positions of the vowels at various stages of the shift. Note that the directions of the axes are reversed. This is standard practice in the sociolinguistics community. In our model, we will use an idealized range of 0-100 for each axis and they will extend in the usual directions.

BOT fronted, we see that BAT continues to shift after BOT has apparently stabilized. We attempt to address this controversy with our model. Second, we see that BOUGHT doesn't shift much at all, contrary to the description of Labov for the Northern Cities region as a whole. Third, BET and BUT show significant movement, and BUT and BOUGHT get quite close to one another. Fourth, though not previously suggested by McCarthy, there could be an interaction between the movements of BAT and BIT. The mechanism of this last movement will remain an open question. It is as yet unaccounted for by our model. We will address all the other aspects, as well as the more general question of the time-scale of the shift, through our model.

Related Work

While the vowel system is computationally well-studied (e.g., Joanisse and Seidenberg, 1997; de Boer, 2001; Dras and Harrison, 2003), there has been relatively little work on modeling vowel *shifts*. What work there has been has focused on different aspects of the phenomenon.

Ettlinger (2007) modeled chain shifts along a single axis (the primary frequency, or f_1) using an exemplar-based model with just two vowels. In his model, the vowel space is divided into Voronoi cells so that any perceived utterance is categorized as the perceptual prototype in the hearer's vowel system to which it is closest. His primary insight is that if the position of a prototype vowel changes for some reason, it will change the boundaries of its Voronoi cell and cause a consequent change in the position of the prototype in adjacent cells (since the prototype is always the centroid of the cell). This is effectively a chain shift. Note that this model does not directly depend on notions like accommodation.

The main problem with this model is that it does not work

in two dimensions. If we include both f_1 and f_2 , then the alterations in the Voronoi diagram can be much more complex when one of the vowels shifts, and it is hard to reproduce exactly the shift that is seen empirically.

Stanford and Kenny (2012) use a similar model, with three vowels, but their focus is different. They address the question of whether there are differences between child vowel acquisition (by *transmission* from parents and others) and adult vowel change (by *diffusion* through interactions between adults). Their model does not examine the mechanisms by which a chain shift happens. They essentially enforce a vowel shift to study the differences due to frequency of contact between agents with different vowel systems.

Lakkaraju et al. (2012) have the model most similar to ours, though they also restrict attention to change along a single axis in a discrete setting. They also use accommodation to account for change, with two constraints: a phonetic differentiation constraint and a total ordering constraint (which says that there is a total ordering on the vowels, preventing them from leapfrogging each other). The present work extends that model in several ways: by considering the full two-dimensional vowel space, in a continuous setting, with arguably more realistic constraints, and a closer comparison with data.

The Model

Convergence in vowel systems

The introduction of a new, incoming linguistic variant such as a shifted vowel creates an opportunity for variation which may ultimately lead to sustained language change as in the NCVS. Linguistic innovation spreads via face-to-face interactions within social networks, and the degree of language change is mediated by a variety of factors, as described below.

According to Communication Accommodation Theory (CAT; Giles and Coupland, 1991), the motivation for convergence for or divergence from their interlocutors is the desire to achieve an optimal degree of social distance. Trudgill (1986), however, proposes that those vowels that are undergoing change in progress might be more salient, and therefore more likely to show accommodation. Goldinger (1998) proposes that convergence may be an automatic cognitive reflex that results from past experience and the type of information that has been stored as exemplars. Under the latter view, accommodation does not depend on social factors.

Babel's (2009) study of accommodation and the California Vowel Shift suggests that different vowels exhibit different degrees of convergence. In an experimental setting, the low vowels (i.e., BAT and BOT/BOUGHT) showed more accommodation in f_1/f_2 values than non-low vowels (i.e., BIT, BOAT, BOOT), perhaps because of the large range of phonetic realizations between stressed and unstressed vowels that result from having to raise and lower the jaw. Social factors (i.e. attitudes and affinity for the ethnic group of the

interlocutor) were shown to mediate the degree of convergence in these experiments.

In Chicago, Herndobler (1977) argues that the spread of raised BAT during the mid-20th Century may have resulted from the perception of urban sophistication associated with the raised variant. This claim would be consistent with socially-motivated accounts of accommodation. More recently, McCarthy (2011) shows that some speakers, especially college-educated ones, have negative stereotypes associated with raised BAT. As social perception of shifted vowels goes from favored to stigmatized, one might expect vowel shifts to peak and retreat. None of the other Chicago vowels implicated in the NCVS appear to be consciously associated with either prestige or stigma.

To summarize, various social and asocial explanations for why speakers converge toward new vowel targets have been offered, and all vowels may not undergo the same amount of accommodation. For the present, we acknowledge but set aside considerations of social perception to focus on the collective dynamics due to transmission across generations.

The Computational Model

We develop an agent-based model of a population undergoing a vowel shift. The population consists of 10,000 agents. Each agent is initially assigned an age between 2 and 70 years. For simplicity, we assume that, as the simulation progresses and the agents' ages increase, agents past the age of 70 die and are replaced with an equal number of agents of age 2, thus keeping the population size constant over time.

New agents begin at age 2 because we abstract away the problem of initial vowel system acquisition, since we are primarily interested in vowel shifting. Vowel system acquisition by infants is non-trivial and has been studied through computational modeling by de Boer (2003). He showed that infants can acquire the vowel system of their parents through a combination of careful articulation by their parents (in child-directed speech), and compensatory expansion of articulations of reduced speech sounds (by the infants). We do not include these complexities in our model, assuming that some such mechanism is present to allow children to acquire the vowel system of their parents in the first two years of their life.

In our model, each new agent is assigned the vowel system of a randomly chosen parent agent. The parent agent is chosen from the subset of the population aged between 21 and 31 years. Adaptation of the vowel system happens between the ages of 2 and 16 years (inclusive). Once an agent reaches 17 years of age, its vowel system becomes fixed.

One simulated year consists of 5,000,000 interactions, where each idealized interaction consists of a *speaker* communicating the position of one of its vowels to a *hearer*. The speaker is chosen from the population a year older than the hearer population. The hearer updates the position of its own corresponding vowel in response, following a sys-

tem of internal constraints explained below. Since agent ages vary from 2 through 70, but learning only happens between the ages of 2 through 16, we have approximately $15/69 * 10000 \approx 2174$ learning agents on average in the population. Since the vowel system consists of 6 vowels, this results in $5000000/(2174 * 6) \approx 383$ updates per vowel per agent per year, which is close to one update per vowel per agent per day.

Updates happen through *accommodation*, which simply means that the hearer tries to move the position of its vowel closer to the perceived position of the speaker's vowel.

The key assumption of our model is that accommodation incorporates representational momentum (Kelly and Freyd, 1987; Freyd et al., 1990). The principle of representational momentum is well-studied in psychology, and states that humans have a forward memory asymmetry for pitch (in the auditory case; it has also been demonstrated for other sensory modalities). When subjects are presented with a sequence of tones of rising pitch, they later recall the pitch of the final tone to be higher than it was. Freyd and her colleagues have also shown that the distance between the remembered final pitch and the actual final pitch is proportional to the implied velocity. Therefore they explain this phenomenon in terms of a momentum effect.

The notion of momentum is also common in artificial neural network learning, where it is used to improve convergence time and reduce oscillations in the weights (e.g., see Haykin, 1998, p. 170).

In the model we include a momentum term in the vowel update as,

$$v_i^{t+1} = v_i^t + (1 - \alpha)\eta(v'_i - v_i^t) + \alpha(v_i^t - v_i^{t-1}), \quad (2)$$

where α is the momentum factor, η is the learning rate, v'_i is the target position for vowel v_i , and t is the time step.

After the new position of the vowel has been calculated in this way, two constraints are applied to decide its final position in time step $t + 1$.

- *Differentiability constraint*: If the speaker's vowel position is too close to the position of an alternate vowel in the hearer's vowel system, the hearer will not accommodate.
- *Margin of security constraint*: If an update brings a vowel too close to another vowel, both vowels get pushed apart.

If two vowels get too close to each other, there is a chance they will *merge*. The differentiability constraint acts to prevent mergers between vowels in the hearer's vowel space, and the margin of security constraint acts to repair the system if two vowels get too close. In reality, mergers do occur, and the relation between mergers and shifts is poorly understood. If mergers occur, language users can rely on other cues such as vowel duration or conversational context to disambiguate meaning. We don't model these aspects here.

The implementation of the differentiability constraint is straightforward. If v_1 is the vowel being communicated, the hearer compares the perceived position of the speaker's v_1 with the positions of the hearer's vowels v_2 through v_6 (i.e., all vowels other than v_1). If the distance to any of these vowels is too small, the hearer does not update its v_1 position, i.e., we just reset v_1^{t+1} to be the same as v_1^t .

The implementation of the margin of security constraint is a little more complicated because we have to decide a direction for each vowel to move when they push each other. For this we rely on a set of principles distilled by Labov.

Labov's principles

Based on a survey of chain shifts in English and other languages, Labov (1994) proposes that there are three universal principles that constrain the possible movements of vowels.

- Principle I: In chain shifts, tense nuclei ("long" vowels, e.g., BOOT, BEET) rise along a peripheral track (regions close the f1/f2 axes).
- Principle II: In chain shifts, lax nuclei ("short" vowels, e.g., BIT, BET) fall along a nonperipheral track.
- Principle III': In chain shifts, tense vowels move to the front along peripheral paths, and lax vowels move to the back along nonperipheral paths. (Note that this is numbered III' because he revised his earlier principle III).

He has applied these principles to the vowel movements in the NCVS as follows. Principle I applies to the fronting and raising of BAT. Principle II applies to the lowering of BIT (and to BET to some extent). Principle III' applies to the fronting of BOT and the backing of BET and BUT.

In our implementation of the margin of security constraint, therefore, if there is a collision (an update brings two vowels too close to each other), they move in the directions suggested by Labov's principles in an attempt to repair the vowel system. Note that the margin of security may not get re-established in a single update. There is no attempt to re-check after the update if the margin of security constraint is still being violated. This check effectively happens the next time the hearer agent again attempts to update the same vowel and notices the constraint violation again. The distance the vowel moves is determined by a push rate, γ , simply as

$$v_i^{t+1'} = v_i^{t+1} + \gamma u_i, \quad (3)$$

where u_i is a unit vector in the direction suggested by Labov's principles for vowel v_i .

Experiments

As mentioned earlier, the population consists of 10,000 agents. We initialize 40% of the agents in a shifted state, i.e., with raised BAT, fronted BOT, and lowered BOUGHT. The rest of the population has the default initial state with low BAT, back BOT and mid-back BOUGHT. The positions

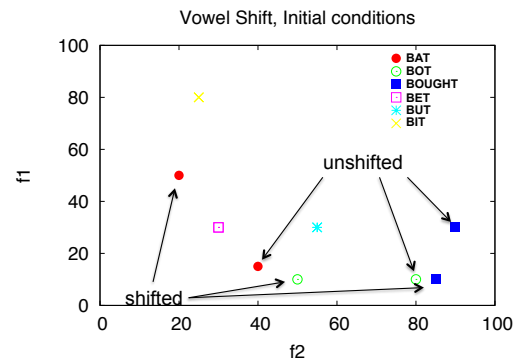


Figure 3: Initial conditions. 40% of the population has the shifted versions of the vowels BAT, BOT, and BOUGHT. The rest have the unshifted versions. The entire population has the same vowel positions for BET, BUT, and BIT.

of the other vowels are the same for all the agents. Both sets of vowels are shown in figure 3.

Parameter settings are as follows: learning rate, $\eta = 0.0001$, momentum, $\alpha = 0.2$, and push rate, $\gamma = 0.00004$. Vowel production is allowed to be noisy, by adding a random variable sampled from a circular Gaussian with zero mean and standard deviation 10.

Noise alone, even if momentum is zero, can cause a partial shift. This is an interesting finding in itself. So for comparison we also ran an experiment where all the settings are exactly the same, except that the momentum factor, α , is set to 0.

We run the simulation for 160 simulated years, and then extract the average vowel positions for age-groups that are 20 years apart.

The results of the experiments, and a statistical comparison of the cases with and without momentum are presented in the next section.

Results and Comparison

The results of the two experiments are plotted in figures 4a and 4b. The figures show the mean position of each vowel for each age groups. The age groups are chosen to be 20 years apart each, giving us eight groups from 160 simulated years. Group 1 corresponds to the subset of the population that was 20 years old at the beginning of the simulation, thus their vowel systems are in the initial condition and don't change. The next group corresponds to the subset that was 20 years old in year 20, followed by the group that was 20 years old in year 40, and so on. In each case, they are sampled once the agents are past the point of adapting their vowel systems.

From figure 4, it is immediately obvious that we see much more shifting in the case with momentum. A clearer comparison of the magnitude of shifting is seen in figure 5,

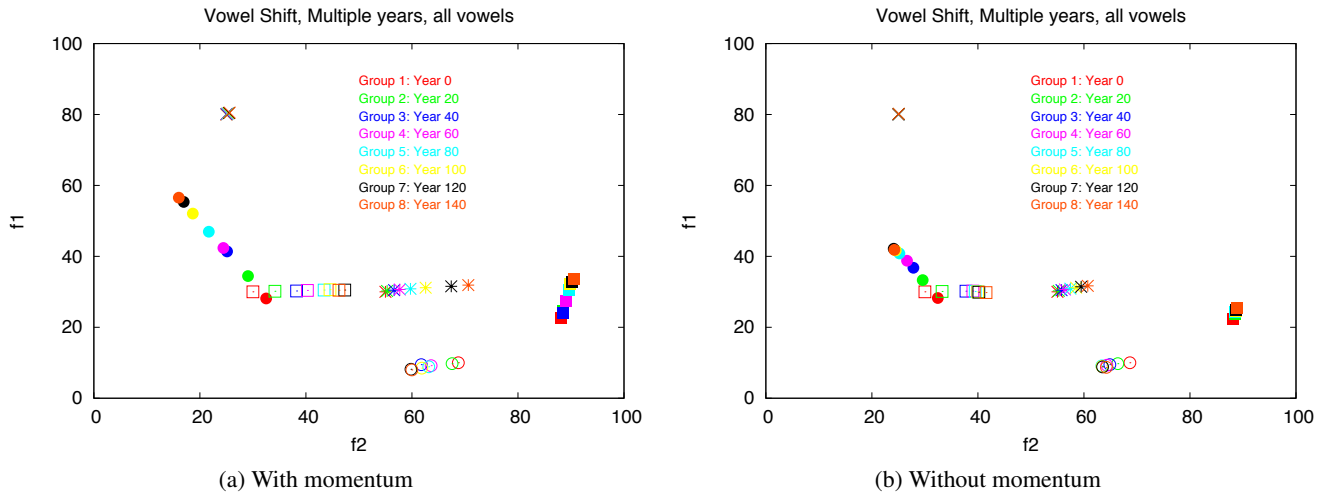


Figure 4: Vowel shifts generated by the model with and without representational momentum. For each age group, we plot the mean position (over that age group) of each vowel. The point shape denotes the vowel and the point color denotes the age group. We see that shifts are much more distinct when the model includes momentum.

which also shows a comparison with the data in figure 2. In each case, the magnitude of the shift is greater with momentum than without. Note that since BOT is fronting, i.e., moving in the direction of decreasing f_2 , the plot in figure 5b shows that the shift with momentum is greater for BOT because the curve for BOT with momentum is below that without momentum.

The comparison with data is done by linearly transforming the empirical data so that the position for BOT from the speaker group born during 1890-1910 lies on top of the position for BOT from group 4 of the simulation with momentum. All the other data points are then transformed using the same mapping. Even this naïve mapping shows a good fit with the simulation results.

BAT and BOUGHT show close overlap between the empirical data and the simulation results with momentum. BUT shows a close match in the slope, i.e. the magnitude of the shift from one generation to the next, though the position suggests that we have chosen the initial position to be too fronted.

BOT shows a weaker match between data and simulation results, but as we will see below, in the case of that vowel, most of the shifts are not statistically significant. BET shows a steeper change in the data than in either simulation, though it is closer to the simulation with momentum.

The largest mismatch between the data and the simulation results for each vowel is for the earliest group, where the sample size is the smallest, consisting of only 3 speakers.

We do not show error bars in the plots to avoid clutter. However, we present a detailed statistical analysis below.

In order to examine change over time, we do a MANOVA to examine the statistical relationship between age group and

vowel position. The partial η^2 values are shown in table 1. This is a measure of the effect size. It tells how much of the variation in the shift of a vowel is attributable to the age group. We see that the values for the experiment with momentum are higher than or close to the values for the experiment without momentum, and are closer to the data.

Table 1: A comparison of the partial η^2 with and without momentum, and from the data. We see that the variation attributable to the group is in general larger for the experiment with momentum, and closer to the data.

Vowel	Partial η^2		
	w. m.	w.o. m.	data
BAT f1	0.276	0.19	0.34
BOT f2	0.045	0.036	0.226
BOUGHT f1	0.207	0.022	0.209
BET f2	0.195	0.207	0.558
BUT f2	0.503	0.559	0.346

As a rule of thumb, values of partial η^2 of 0.1 or below are considered small, values around 0.2 are considered medium, and anything over 0.3 is considered a fairly big effect. The largest effects are seen for BUT, and for BAT with momentum. BOUGHT with momentum shows a significantly larger effect than without momentum.

We also do post-hoc tests between the age groups for each vowel, with and without momentum, and from the data. We don't have the space here to present the entire set of p-values, but the results can be summarized as follows.

- For the movement of BAT with momentum, all the pairwise tests are significant at the $p < 0.05$ level except for groups 3 and 4, and groups 6, 7, and 8. Without momen-

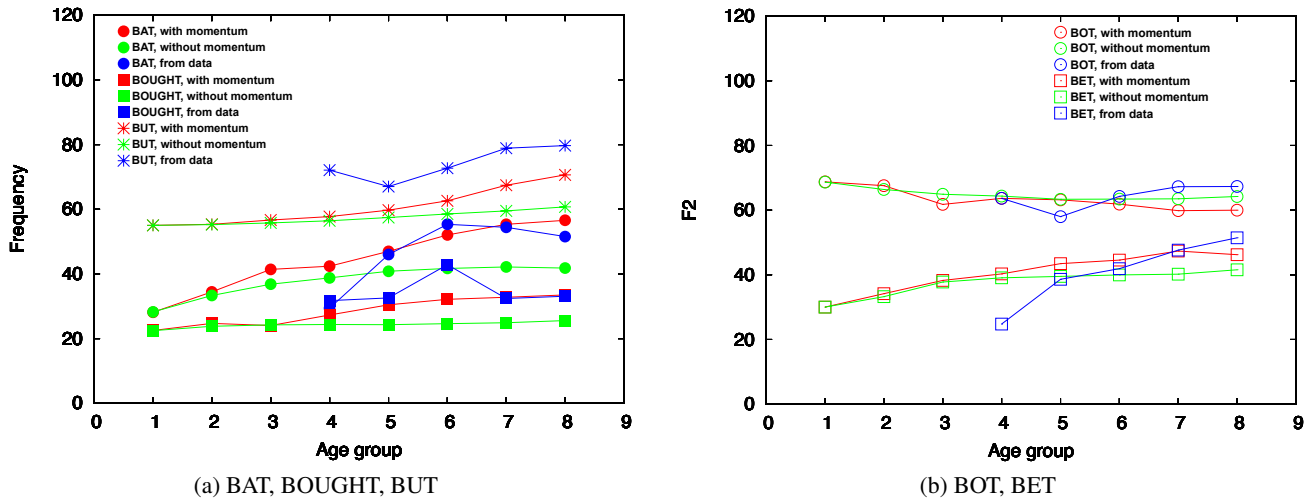


Figure 5: A comparison of the shift magnitude with and without momentum, and with the transformed empirical data. Only movement along the significant axis for each of the five vowels of interest is shown. The axes of interest are f1 for BAT and BOUGHT, and f2 for BOT, BET, and BUT. The comparison for BET is plotted along with BAT and BOUGHT to avoid overlap.

tum, the shifts are not significant from group 5 onwards. In the data, the shifts of BAT are only significant between the first group (1890-1910) and the rest.

- For the movement of BOT with momentum, the shifts are not significant from group 3 onwards, while without momentum, they are not significant from group 2 onwards. In the data, the shifts of BOT are not significant.
- For the movement of BOUGHT with momentum, the shifts are not significant early and late, i.e., between groups 1, 2, and 3, and from group 5 onwards. Most of the significant shifting happens from groups 3 to 4 to 5, though 5 vs. 8 is also significant ($p < 0.005$). Without momentum, none of the movements are significant at $p < 0.05$, except 1 vs. 8. In the data, the shifts of BOUGHT are not significant.
- For the movement of BET with momentum, the shifts are not significant from group 5 onwards, while they lose significance from group 3 onwards without momentum. In the data, the shifts are significant when comparing groups 1 and 2 with the rest, but not amongst the rest.
- For the movement of BUT with and without momentum, the shifts remain significant throughout except for groups 7 vs. 8 in the case with momentum. In the data, the shift between group 2 and 5 is significant, but the others aren't.

Discussion

To fully comprehend the results, we have to consider both the magnitude and the significance of the shifts. Taken together, several interesting inferences can be drawn from the results.

First, overall, shifts with momentum are larger, and continue for longer, than shifts without momentum. Shifts with momentum naturally come to an end on a time-scale of about 100-140 years, as the overshoot due to momentum dies out when movements get smaller, which matches well with observations. The magnitudes of the shifts in the data match quite well with the simulation with momentum.

More specifically, we see that BAT and BOT start moving more or less simultaneously, but BOT completes its movement early, while the movement of BAT slows down between groups 3 and 4 but gets a boost due to the push interaction with BET. This suggests a possible resolution to the ordering controversy between BAT and BOT mentioned earlier. McCarthy (2009) has suggested, based on her data and contrary to the chronology suggested by Labov, that the ordering is not simply BAT fronting and raising followed by BOT fronting, but rather that the movements might be interleaved, with BAT fronting followed by BOT fronting followed by BAT raising. However, it was not known what might cause such interleaving.

Our simulation is supportive of McCarthy's chronology where we see BAT raising and fronting until it comes near BET, followed by BOT fronting, followed by the second stage of BAT raising and fronting due to the push interaction with BET. This suggests the push interaction as a possible cause for the interleaving.

The movement of BOUGHT with momentum is actually upward, contrary to Labov's suggestion, but matching McCarthy's data (fig. 2) where the position of BOUGHT is close to the initial position or even higher. Without momentum BOUGHT ends up in a position in-between the initial shifted and unshifted positions, which is unrealistic.

BET and BUT show strongly significant movement, both with and without momentum, but as can be seen from both fig. 4 and fig. 5, the magnitude of the shift is much greater with momentum. Without momentum, the groups bunch up very quickly, resulting in only a partial shift.

Conclusion

We have presented an agent-based model of vowel shifts based on empirical principles that have been derived in the sociolinguistic community. We have shown that an additional ingredient, viz. representational momentum, is required to explain several aspects of the NCVS in Chicago.

The main missing piece is an account of the movement of BIT, which shows robust movement in Chicago but relatively little in our model. The data in fig. 2 suggests a possible interaction between BAT and BIT, in that BAT raising may push BIT backward in accordance with Principle III'. Alternatively, BIT's movement could be due to its involvement in a parallel shift, in which front lax vowels BET and BIT move backward as a class. We have not implemented this principle, although it may be worth considering.

More generally, we need a deeper account of how vowels move when they come too close to each other. Labov's principles are essentially *ad hoc* rules derived from observation. A cognitive and psychoacoustic perspective might provide an energy-function based approach to vowel spacing and interaction. Another approach might be based on neural coding, similar to Joanisse and Seidenberg (1997).

Another direction in which this work can be extended is to incorporate social and economic demographic-based variation, which might help explain the regional variations in the NCVS and in other shifts.

In conclusion, we believe that computational simulation has much to offer the study of vowel shifts and other large-scale dynamical phenomena in language because through simulation we can shed light on precisely those questions for which the available data are sparse and hard to gather.

Acknowledgements

S. S. was supported in part by DTRA R&D Grant HDTRA1-09-1-0017, DTRA CNIMS Contract HDTRA1-11-D-0016-0001, NSF NetSE Grant CNS-1011769, NSF HSD Grant SES-0729441 and NSF PetaApps Grant OCI-0904844. C. M. was supported in part by the GMU College of Humanities and Social Sciences.

References

- Babel, M. (2009). *Phonetic and Social Selectivity in Phonetic Accommodation*. PhD thesis, Dept. of Linguistics, University of California, Berkeley.
- Beckner, C., Blythe, R., Bybee, J., Christiansen, M. H., Croft, W., Ellis, N. C., Holland, J., Ke, J., Larsen-Freeman, D., and Schoenemann, T. (2009). Language is a complex adaptive system: Position paper. *Language Learning*, 59:1–26.
- de Boer, B. (2001). *The Origins of Vowel Systems*. Oxford University Press.
- de Boer, B. (2003). Conditions for stable vowel systems in a population. In *ECAL03*, pages 415–424.
- Dras, M. and Harrison, K. D. (2003). Emergent behavior in phonological pattern change. In Standish, Abbass, and Bedau, editors, *Proceedings of the 8th International Conference on Artificial Life*, pages 390–393. MIT Press.
- Ettlinger, M. (2007). An exemplar-based model of chain shifts. In *Proceedings of the 16th International Congress of Phonetic Sciences*, pages 685–688, Saarbrücken, Germany.
- Freyd, J. J., Kelly, M. H., and DeKay, M. L. (1990). Representational momentum in memory for pitch. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16(6):1107–1117.
- Giles, H. and Coupland, N. (1991). *Language: Contexts and Consequences*. Open University Press.
- Goldinger, S. D. (1998). Echoes of echoes? An episodic theory of lexical access. *Psychological Review*, 105(2):251–279.
- Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation* (2nd ed.). Prentice Hall.
- Herndobler, R. (1977). *White Working-Class Speech: The East Side of Chicago*. PhD thesis, University of Chicago.
- Joanisse, M. F. and Seidenberg, M. S. (1997). [i e a u] and sometimes [o]: Perceptual and computational constraints on vowel inventories. In *Proceedings of the 15th Annual Conference of the Cognitive Science Society*.
- Kelly, M. H. and Freyd, J. J. (1987). Explorations of representational momentum. *Cognitive Psychology*, 19:369–401.
- Labov, W. (1994). *Principles of Linguistic Change Vol. 1: Internal Factors*. Blackwell, Cambridge, MA.
- Labov, W., Ash, S., and Boberg, C. (2006). *The Atlas of North American English: Phonetics, Phonology, and Sound Change*. Mouton de Gruyter, Berlin.
- Lakkaraju, K., Swarup, S., and Gasser, L. (2012). Consensus under constraints: Modeling the Great English Vowel Shift. In *The International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction (SBP)*, College Park, MD.
- Martinet, A. (1955). *Economie des changements phonétiques*. Francke, Bern, Switzerland.
- McCarthy, C. (2009). The Northern Cities Shift in real time: Evidence from Chicago. *Penn Working Papers in Linguistics*, 15(2):article 12.
- McCarthy, C. (2011). The Northern Cities Shift in Chicago. *Journal of English Linguistics*, 39(2):166–187.
- Stanford, J. N. and Kenny, L. A. (2012). An agent-based computational interpretation of the transmission and diffusion of vowel chain shifts across large communities. *Unpublished manuscript*.
- Thomas, E. (2001). *An Acoustic Analysis of Vowel Variation in New World English*. Duke University Press, Durham, NC.
- Thomas, E. R. and Kendall, T. (2007). NORM: The vowel normalization and plotting suite. Online Resource: <http://ncslaap.lib.ncsu.edu/tools/norm/>.
- Trudgill, P. (1986). *Dialects in Contact*. Basil Blackwell, Oxford.

Environment classification in multiagent systems inspired by the adaptive immune system

Danesh Tarapore^{1,2}, Anders Lyhne Christensen³, Pedro U. Lima¹ and Jorge Carneiro²

¹Institute for Systems and Robotics (ISR), Instituto Superior Técnico (IST), Lisbon, Portugal

²Instituto Gulbenkian de Ciência, Oeiras, Portugal

³Instituto de Telecomunicações & Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal
daneshatarapore@gmail.com

Abstract

The adaptive immune system in vertebrates is a complex, distributed, adaptive system capable of effecting collective multicellular responses. Our study introduces many of the desirable properties of this biological system to decentralized multiagent systems. We adopt the crossregulation model of the adaptive immune system involving interactions between effector and regulatory cells. Effector cells can mount beneficial immune responses to microbial antigens as well as pathologic autoimmune responses to self-antigens. Deleterious autoimmunity is prevented by regulatory cells that suppress the effectors to tolerate the self-antigens. We redeploy the cross-regulation model within a multiagent system by letting each agent run an ODE-based instance of the model. Results of extensive simulation-based experiments demonstrate that a distributed multiagent system can mount different responses to distinct objects in their environment. These responses are solely a result of the dynamics between virtual cells in each agent and interactions between neighboring agents. The collective dynamics gives rise to a meaningful “self”-“nonself” classification of the environment by individual agent, even if these categories were not prescribed a priori in the agents.

Introduction

Multiagent systems (MAS) comprise a large number of research domains, ranging from software agents to multirobot systems, and play an important role in several applications, such as supply chain management, transportation logistics and network routing. The coordination of agents in a MAS is a major challenge because agent behavior depends not only on interactions with their immediate environment but also on the behavior of other agents. A centralized control approach may not always be feasible due to computational and/or communication constraints on agents (e.g., Crespi et al. (2008); Mermoud et al. (2010)). Distributed control, on the other hand, is often complicated to realize because the behavioral rules for the individual units cannot be easily derived from a desired macroscopic behavior (e.g., Parker (2000); Yamins and Nagpal (2008); Hamann (2010)). In the design of large scale distributed systems, several researchers have therefore taken inspiration from nature e.g., aggregation of amoeba into slime mold (Payton et al., 2003), quorum sensing and communication in bacteria (Sahin, 2005),

division of labor in social insects such as ants and honey bees (Parker et al., 2003; Waibel et al., 2009; Hauert et al., 2009; Tarapore et al., 2010; O’Grady et al., 2010).

The cell collective that constitutes the adaptive immune system has been extremely successful during the course of evolution as evidenced by its presence in all jawed vertebrate species (Janeway et al., 1997). Central to the success of these cells is the important role they play in establishing and maximizing the capabilities of the immune system, by allowing an exquisite “self-nonself” discrimination that is not present in invertebrates. The cell collective is able to recognize and mount specific immune responses to microbial agents that the organism and its ancestors had never faced before. It does this immersed in the constant presence of diverse and abundant body antigens, which are molecularly similar to the microbial antigens. In normal healthy individuals, sporadic microbial invaders are specifically eliminated by immune responses and, at the same time, pathologic autoimmune responses to the abundant body antigens is prevented, i.e. natural tolerance to “self” is maintained. Experimental evidence indicates that natural tolerance results from the dynamics and interactions between specific regulatory and effector T-cells (e.g., Sakaguchi (2004)). Interestingly, the decentralized nature of the interactions may impart a high degree of robustness for natural tolerance, without the need of maintaining a specific, genetically hardwired, “memory” of self-antigens.

The decentralized and adaptive nature of the immune system is a source of inspiration for designers of large scale MAS. In particular, the ability of the system to dynamically maintain natural tolerance has many industrial applications. Some typical studies that take inspiration from this “self”-“nonself” discrimination capability of the immune system include, distributed intrusion detection systems (Nino and Beltran, 2002; Kim and Bentley, 1999), and fault tolerance systems (Bradley and Tyrrell, 2000, 2001; Canham and Tyrrell, 2002). However, most of these models assume which particular antigens or features are prescribed as “self”, and consequently the system is trained to tolerate them. While this approach does provide some in-

interesting results of robust feature classification, it does not fully incorporate the dynamics and adaptive nature of the immune system. This led us to propose the use of the *cross-regulation model* (CRM) for the maintenance of tolerance. The CRM (Leon et al., 2000, 2003, 2004; Carneiro et al., 2007) suggests a dynamics of interactions between cells of the immune system, that allows the system to discriminate between antigens based solely on their density and persistence in the environment. The system is able to tolerate body antigens (i.e. “self”) that are characteristically persistent and abundant, and to mount an immune response to foreign pathogens, that are characterized as being neither persistent nor abundant. The model has been used successfully in spam detection (e.g., Abi-Haidar and Rocha (2008)) and document classification (e.g., Abi-Haidar and Rocha (2010, 2011)) scenarios, making it a good candidate for MAS for environment classification.

In this study, we propose a CRM-based approach to replicate the capability of the immune system in maintaining tolerance. We use an agent-based simulator to model a situation where individuals have to tolerate certain features, while mounting an immune response against others. The different environmental features are represented by different sensory stimuli in the environment, and their nature (“self” or “nonself”) are not known by the agents beforehand. We demonstrate the capacity of the system to tolerate specific environmental features that may be characterized as persistent and abundant (“self”), while mounting an immune response against others (“nonself”). In addition, the system response is resilient to sensory noise, and can respond correctly under varying environmental conditions.

The rest of the paper is organized as follows: In the following section, we describe the CRM. We then present the application of the CRM in a MAS. We go on to report the results of our experiments in different environmental conditions and under varying levels of perceptual noise. Finally, we discuss our approach to environment classification and highlight the conclusions of this study.

The Crossregulation Model

Two general principles are essential for the viability of multicellular organisms. Firstly, the persistence of any cell lineage requires that its cells recurrently interact with other cell types in the organism. Cells that fail to interact with other cells eventually die. Secondly, the growth of a cell population involves density-dependent feedback mechanisms controlling individual cell proliferation. These feedback mechanisms may involve (i) indirect interactions among cells (such as a competition for limited growth factors) and (ii) direct interactions, such as contact inhibition. These two principles of multicellular organization are the foundation of the crossregulation model, and have been justified extensively in Carneiro et al. (2007). Below, we outline the model and highlight its interesting properties that are later replicated

with a cell recruitment mechanism.

The CRM describes the population dynamics of cells of the adaptive immune system, based on three mutually interacting cell types: (i) Antigen presenting cells (APCs) that display the antigen on their surface. Individual APCs have a fixed number of sites (s) on which effector and regulatory cells can form conjugates; (ii) effector cells T_E that can potentially mount an immune response which, depending on receptor specificity, can be directed to foreign pathogens or to self-antigens; and (iii) regulatory cells T_R that suppress proliferation of T_E cells with similar specificities. Furthermore, the APCs are classified into different sub-populations of equivalent APCs, with each APC in a sub-population presenting the same antigen on its surface. Similarly effector and regulatory cells are also classified into different sub-populations or clones according to their specificity.

The dynamics of T-cell population is regulated by the following density-dependent feedback mechanisms. (i) Effector and regulatory cells that are unable to interact with APCs are slowly lost by cell death. (ii) The proliferation of effector and regulatory cells requires interactions with APCs and depends on interactions these T-cells make with each other. Proliferation of the T_E cell population is promoted by the absence of regulatory cells on the APC. In contrast, T_R can only proliferate following co-conjugation with effector cells on the same APC. Additionally, T_E and T_R cells interact indirectly by competition for access to conjugation sites on APCs.

Behavior of cell population

Considerable work has focused on analyzing the properties of the CRM, and the underlying dynamics between T_E , T_R and APCs (Leon et al., 2000, 2003). An interesting characteristic of the CRM is the ability to discriminate between antigens based on their density. At low concentrations of APCs, the system evolves into a stable state composed only of effector cells (immune response). In contrast, at higher values of APCs, the system demonstrates bistable behavior. At these concentrations of antigens, the system can evolve either into an equilibrium state consisting predominantly of effector cells (immune response), or into a state composed largely of regulatory cells (tolerant response). The system develops into the regulatory cell dominated state, provided that the seeding population has sufficient T_R cells. By contrast, if T_R cells are initially underrepresented, T_E cells will competitively exclude the former from the system. Consequent to the antigen density dependent response, the effector cells are made tolerant to antigens that are persistent and abundant. In addition, the effector cells are free to mount immune responses to antigens that are not persistent or not abundant.

Table 1: Parameters of the crossregulation model.

Param.	Description	Value (a.u.)
A_j	Density of APCs of population j	—
s	Maximum number of T-cells that can bind to an APC	3
E_0	Seed density of effector cells	10
R_0	Seed density of regulatory cells	100
E_i	Density of effector cells of clone i	—
R_i	Density of regulatory cells of clone i	—
T_i	Density of T-cells of clone i	$E_i + R_i$
C_{ij}	Density of conjugates between T_i and A_j	—
γ_c	Conjugation rate of T-cells to APCs	10^{-1}
γ_d	Deconjugation rate of T-cells from APCs	10^{-1}
σ_E	Influx rate of new effector cells	10^{-3}
σ_R	Influx rate of new regulatory cells	0.6×10^{-3}
π_E	Proliferation rate of effector cells	10^{-3}
π_R	Proliferation rate of regulatory cells	0.5×10^{-3}
δ	Death rate of effector and regulatory cells	10^{-5}

Mathematical formulation of the model

The dynamics of the interactions between effector and regulatory cells, with APCs is described by a set of ordinary differential equations in the following variables: (i) The number of effector E_i and regulatory R_i T-cells of clonal type i , where $i \in \{1, 2, \dots, N\}$ and N is the number of T-cell clones. (ii) The number of APCs A_j , where $j \in \{1, 2, \dots, M\}$ and M is the number of different antigen types. (iii) The number of conjugates C_{ij} formed between effector and regulatory cells from clone i and APC from population j .

For the effector E_i and regulatory R_i cells of clone i , we have:

$$\frac{dE_i}{dt} = \sigma_E + \pi_E E_i^* - \delta E_i \quad (1)$$

$$\frac{dR_i}{dt} = \sigma_R + \pi_R R_i^* - \delta R_i \quad (2)$$

where the involved quantities are defined in Table 1.

The equations for E_i (eq 1) and R_i (eq 2) have three terms. The first term represents the influx of new cells, which is assumed to be constant. The second term accounts for the proliferation of activated effector and regulatory cells. Finally, the death of T-cells is represented by the third term of the equations. In the simulations, we generate all T-cell clones with similar initial conditions i.e., $\forall i$, $E_i(0) = E_0$ and $R_i(0) = R_0$.

The density of activated T_E and T_R cells of each clone are computed in a stepwise manner. Let us consider the interactions between the i -th T-cell clone and the j -th APC population. The dynamics of the conjugates C_{ij} is described by the following equation:

$$\frac{dC_{ij}}{dt} = \gamma_c \theta_{ij} \left(T_i - \sum_{j=1}^M C_{ij} \right) \left(A_j s - \sum_{i=1}^N C_{ij} \right) - \gamma_d C_{ij}$$

where $T_i = E_i + R_i$, and γ_c and γ_d involve the conjugation and deconjugation rates between APCs and T-cells, respectively (parameters in Table 1). In the above equation, new conjugates are formed by the free T-cells of clone i with the available sites on APCs of population j at rate γ_c . The conjugation rate is also controlled by the affinity (θ_{ij}) between the T-cells and APCs. The existing conjugates dissociate at rate γ_d . The conjugation and deconjugation of T-cells from the APCs is a fast process with respect to the overall T-cell clone dynamics. Consequently, we solve at each time step, the steady state values of the conjugates by the Euler-Heun adaptive step method (Butcher, 2003).

The density of activated effector E_i^* and regulatory R_i^* cells can now be calculated (for details see Appendix A). Conjugated effector cells are activated in the absence of regulatory cells on the same APC. In contrast, conjugated regulatory cells can only be activated if at least one effector cell is simultaneously conjugated to the same APC.

The population dynamics behavior exhibited by the CRM is governed by two key composite parameters representing the effective growth rates of T_E and T_R cell populations (Leon et al., 2000). These two parameters are directly proportional to the basic parameters controlling population growth i.e., conjugation constant (γ_c), affinity between T-cell and APCs (θ_{ij}), influx rate of new effector and regulatory cells (σ_E and σ_R), proliferation rates of these two types of T-cells (π_E and π_R), and the density of APCs (A_j). The effective growth rates of the T-cells is also inversely proportional to the death rate (δ) of the corresponding population. The composite T_E and T_R growth parameters define four parameter regimes according to the resulting cell population behavior. Three parameter regimes result in a single stable state that may correspond to either: (i) extinction of all T-cells ($T_E = 0, T_R = 0$), (ii) immune state ($T_E > T_R$), or (iii) tolerant state ($T_E < T_R$). The fourth parameter regime corresponds to a bistable system where both immune and tolerant states are stable. A detailed analysis of these parameter regimes is provided in Leon et al. (2000). For our present study, the parameter values have been set so that at low APC densities, the system evolves into a single state composed only of effector cells. By contrast, at relatively high density of APCs, the system is bistable and can evolve either into an immune or tolerant equilibrium state.

CRM in a Multiagent System

In this section, we demonstrate how the CRM can be implemented on a distributed embodied multiagent system in order to give the system the capacity to classify different features in the environment based on their concentrations. Features that are persistent and abundant are to be tolerated, while features that are present at a low density are not. We show that the multiagent system is able to adapt online and that it is resilient to perceptual noise.

We use a stochastic, spatial, discrete-time simulator. The

simulated environment is toroidal and has a size of 10×10 units. The MAS is composed of 50 point-sized agents that perform a random walk: each agent move at a constant speed of 0.01 units/time-step, and has a probability of 0.01 of changing to a new random direction each simulation step. The agents detect features of static objects within their sensory range (1 unit) and run an internal and individual instance of a CRM in order to determine if the objects should be tolerated or not (see details below).

Individual features of the static objects in the environment are encoded in Boolean form (present= 1, absent= 0), and then concatenated to form a binary string, the *feature vector*. At the start of each time-step, an agent computes the density of each feature vector (FV_j) within its sensory range. In the agent's internal CRM instance, APCs are then generated corresponding to each of the feature vectors perceived. Each APC presents an individual feature vector to the T-cells. The number of each type of the APCs generated $APC_j = FV_j$, for $j \in \{1, \dots, M\}$, where M is the number of different feature vectors perceived by the agent.

The T-cell clones (T_1, T_2, \dots, T_N), each have a different receptor encoded as a binary string, which determines their affinity to the APC population. The affinity between T cell clonal i and APC population j is denoted by θ_{ij} :

$$\theta_{ij} = \exp\left(-\frac{H(i, j)}{c}\right) \quad (3)$$

where H is the Hamming distance between the receptor of T_i and the feature vector presented by A_j , and c is the cross-reactivity between T-cells and APCs. A high value of c would result in all T-cell clones having a high affinity to all APC populations. By contrast, at low c , each T-cell clone would have a high affinity to only one distinct APC population.

At the start of the simulation, the number of effector and regulator cells on each agent is initialized to E_0 and R_0 respectively. Following this, Algorithm 1 (parameters in Table 2) is performed by the agents in each simulation time-step, allowing the agents to execute the behavior designed in the CRM. The agents begin by sensing their local environment and computing the density of feature vectors. Perceptual noise is modeled by randomly flipping the binary representation of one of the feature with probability x . The CRM is then numerically integrated for time S , allowing the system to respond to the different APCs. After computing the number of effector and regulatory cells at time S , the cells diffuse among agents. In this communication phase, each agent selects a neighboring agent within its communication range. The selection is random following a linear distribution on the total number of T-cells associated with each agent in communication range. Following the selection, each agent sends and receives d of its effector and regulatory cells. Finally, the agent decides the nature of each feature vector FV_j sensed, as follows:

Table 2: Parameters of the stochastic simulator

Param.	Description	Value (a.u.)
N	Number of T-cell clones	4
M	Number of different feature vectors	4
c	Cross-reactivity between T-cells and APCs	0.4
x	Probability to add noise on a feature	0.1 – 0.5
S	Time CRM instance is executed, in a single simulation-step	10^5
d	Proportion of T-cells diffused to neighboring agents	0.5

$$E = \sum_{i=1}^N \theta_{ij} E_i \quad R = \sum_{i=1}^N \theta_{ij} R_i$$

where the feature vector is accepted as tolerant if $R > E$, else the object associated with the feature vector is removed from the environment by the agent.

Algorithm 1 An agent's control loop (simulation of an CRM instance)

- 1: {Perceive static objects}
 - 2: Compute density of feature vectors (FV_j) in sensory range of agent
 - 3: For each of the sensed feature vectors, add noise to one of the features with probability x
 - 4: Assign feature vectors to APCs i.e., $\forall j, A_j = FV_j$
 - 5: {Run instance of CRM}
 - 6: $time \leftarrow 0$
 - 7: **while** $time \leq S$ **do**
 - 8: $\forall i \in \{1, 2 \dots N\}$ and $\forall j \in \{1, 2 \dots M\}$, compute the number of conjugated cells C_{ij} in steady state, integrating using the Euler-Heun adaptive step method
 - 9: Using the number of conjugated cells, compute the updated number of effector and regulatory cells with the Euler-Heun adaptive step method. The adaptive step size is stored in h
 - 10: $time \leftarrow time + h$
 - 11: **end while**
 - 12: {Diffuse cells across neighboring agents}
 - 13: Randomly select one of the agents in the communication range following a linear distribution and weighted by the total number of cells on the respective neighboring agents
 - 14: Exchange cells with agent
 - 15: {Decide if feature vectors are to be tolerated or not}
 - 16: For each feature vector, compute the sum of effector and regulatory cells, weighted by their affinity.
 - 17: Tolerate the FV if total regulatory cells exceeds effectors, else mount an immune response i.e., remove the static object associated with the feature vector from the environment.
-

Experiments

We set up a series of experiments in order to evaluate the classification capabilities of a multiagent system operating according to the model described above. In a first set of experiments, we distributed two different types of static objects in the environment: one with a high density ($10/\text{unit}^2$) and one with a low density ($1/\text{unit}^2$). Both types of static objects were placed at random positions drawn from a uniform distribution. In each replication of the experiment, the feature vectors of the two types of static objects were picked at

random in such a way that one would be the complement of the other. Within the CRM conceptual framework the abundant objects are interpreted as body/self-antigens, while the low density objects are foreign or “nonself”. We endowed agents with the capacity to remove objects and therefore tolerance to “self” was interpreted as the persistence of the objects. We show that the MAS is, under some specific non-trivial conditions, able to tolerate abundant objects that will persist and to remove less abundant objects.

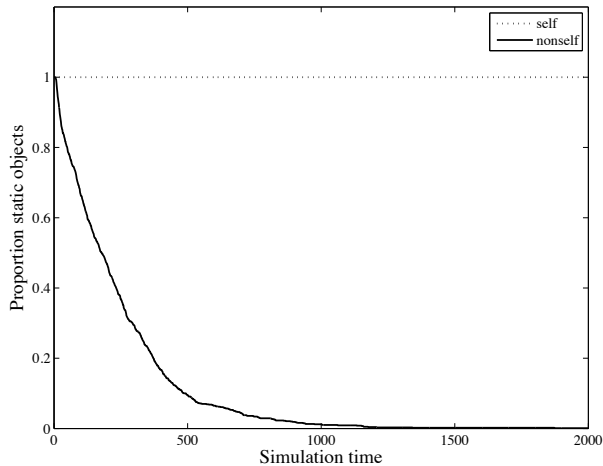


Figure 1: Mean proportion of static objects across 10 replicates. Individual agents had 30% probability perceptual noise.

In Fig. 1, we have plotted the mean proportions of static objects (with respect to the initial quantities) across 10 replicates, each for 2000 simulation steps and with 50 agents in the environment. The object density variance across simulation time was similar irrespective of the level of perceptual noise added ($x = 0.1 - 0.5$), and was therefore illustrated for a single case ($x = 0.3$, Fig. 1). After 2000 simulation time-steps, there was very little variation in the objects associated with “self”. The density of self-objects remaining at 10 for $0.1 - 0.4$ probability of perceptual noise, while at higher level of perceptual noise ($x = 0.5$), tolerance was maintained in all but two replicates (less than 0.2% of self-objects destroyed in each replicate). By contrast, the system exhibited an absence of tolerance to objects associated with “nonself” (Fig. 1 and 2). An immune response to these objects was mounted irrespective of the level of noise. However, the response was more effective at lower levels of noise (Fig. 2).

We set up a second series of experiments in order to evaluate the capabilities of a multiagent system to maintain tolerance under varying environmental conditions. These experiments were designed to assess the requirement for communication between agents. In this set of experiments, we divided the environment into three regions, with two con-

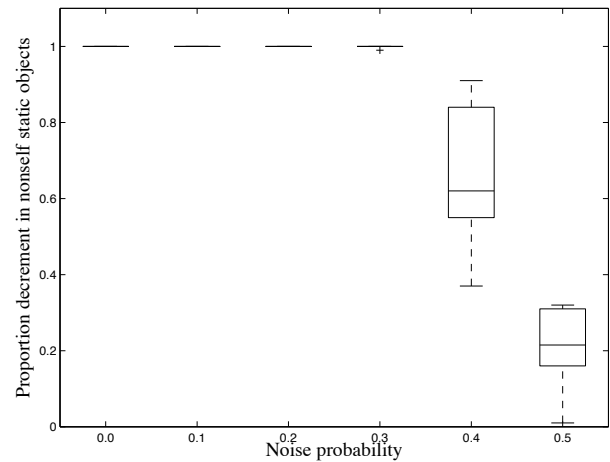


Figure 2: Proportion decrement of “nonself” static objects with different amounts of perceptual noise.

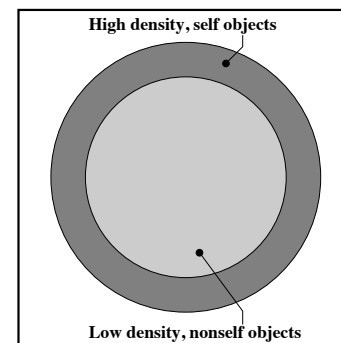


Figure 3: The heterogeneous environment used to investigate environment classification under varying environmental conditions.

centric circle of radii 4 and 5 units (Fig 3). Two different types of static objects were distributed in the environment in two different locations: one with a low density ($1.98/\text{unit}^2$) was distributed within the inner circle, and one with a high density ($70.7/\text{unit}^2$) was distributed between the inner and outer circles.

In Fig. 4 and 5, we have plotted the mean proportions of static objects (with respect to the initial quantities) with intra-agent communication suppressed and enabled respectively. Experiments were replicated 10 times, each for 2000 simulation steps and with 50 agent in the environment.

The communication of T-cells between agents had a strong effect on the maintenance of tolerance. In the absence of communication, the system was unable to maintain tolerance (Fig. 4). At 2000 simulation time-steps, the abundant “self” objects were removed from the environment in all 10 replicates. By contrast, in the presence of communication between agents, almost 100% of abundant “self”

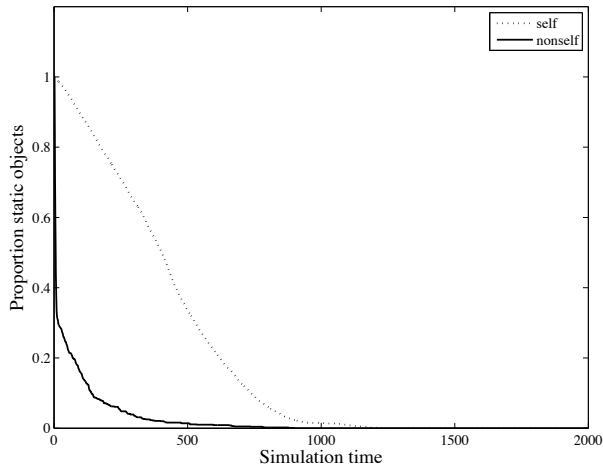


Figure 4: Mean proportion of static objects across 10 replicates, with heterogeneous distribution of static objects (“self” and “nonself”), and inter-agent communication suppressed.

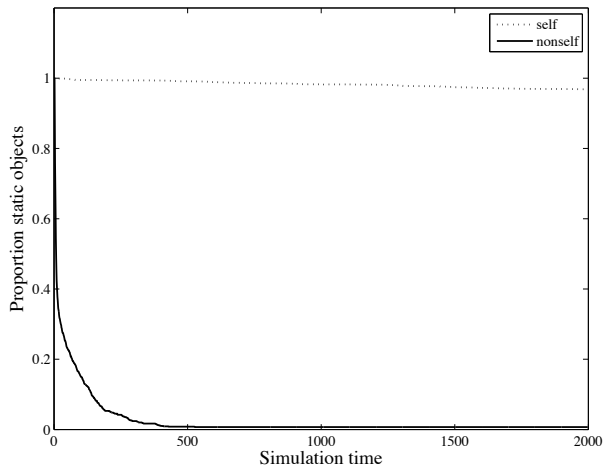


Figure 5: Mean proportion of static objects across 10 replicates, with heterogeneous distribution of static objects (“self” and “nonself”), and inter-agent communication enabled.

objects persisted in the environment at the end of the simulation (Fig. 5). In addition, the agents were also able to mount an effective immune response, such that all the “nonself” objects had been removed from the environment at the end of the simulation (Fig. 5).

Discussion

Our study revealed a robust maintenance of tolerance to “self”, understood as abundant antigens or features, irrespective of the level of perceptual noise on individual agents. Interestingly, even at a 50% chance to distort a sensed fea-

ture, the abundant “self” was largely tolerated. This resiliency to noise exhibited by the system was a consequence of the cross-reactivity between T-cells and APCs. At our level of cross-reactivity, regulatory cells with a high affinity to feature vectors of the “self”, were able to react with and consequently suppress effectors associated with a mis-read “self” feature vector (low Hamming distance apart) and consequently prevent their destruction. Separate experiments investigating the influence of this parameter, indicated a complete absence of tolerance at low values of cross-reactivity. By contrast, at very high levels of cross reactivity, regulatory cells suppressed effectors associated with all the sensed features thus preventing any discrimination by the system. Interestingly, the ability of our system to tolerate noise distinguishes it from a simple response threshold-based model for environment classification, wherein different feature vectors are assigned distinct tolerance thresholds, and the system response is governed strictly by the density of each feature vector type being above or below its corresponding threshold.

In simulated environments with a heterogeneous distribution of objects, the agents continued to classify environmental features correctly, despite the variations in their local environmental conditions. Our results revealed the requirement of communication of T-cells between neighboring agents in order to maintain the tolerance to abundant “self” objects. In the absence of communication, agents were unable to tolerate “self” objects when entering regions consisting of them. By contrast, in the presence of communication, regulatory cells communicated from agents already in the “self” associated region allowed the entering agents to respond faster to environmental changes and consequently, greatly improved their tolerance. The diffusion of T-cells between agents allows the agents to share information of their local environments and to perform better as a collective.

In our simulations, APCs are generated corresponding to each of the feature vectors. Each APC presents an independent feature vector present at that instance. Consequently, APCs related to a newly generated feature vector may not react to the existing T-cells in the agents’ history. This is because the reaction would be dependent on the feature vector chosen for this new event and its affinity to the existing T-cells. We illustrate this point with the following example: Consider an agent in an environment with FV_j presented by A_j at a density resulting in a tolerant response. The agent has in its history, T clonal-type T_i with $\Theta_{ij} = 1$. Consequent to the density of A_j , $R_i > E_i$. Now let us consider the agent moving into an environment resulting in another APC type A_k . However, the existing cells in the agents’ history may or may not react to this new APC, and the decision is stochastic and dependent on the choice of the new feature vector FV_k . In this system, for the existing T-cells to react with the new feature vector, $\theta_{ik} > 0$ and this is a direct consequence of the (preexisting) affinity mapping between

feature vectors and T-cell clonal types. Another possible approach wherein the history of the system could be explicitly taken into account would be the generation of APCs to represent various combinations of feature vectors. Based on the above example, APCs would present feature vectors of type $\{FV_j, FV_k, FV_j FV_k\}$. In this condition, existing cells in the agents' history would be able to respond to new feature vectors. Additionally, the system response would not solely be a consequence of the feature vector specific topology. However, the outcome of this scenario needs to be explored further.

In our experiments, we used a relatively abstract stochastic simulation in which mobile agents performing random walk perceived features on static objects present in the environment. The agents task, to distinguish between what is persistent and abundant and what is not, is a metaphor for a large class of detection and identification tasks in the field of MAS, and more specifically in multirobot systems (MRS): novelty detection, fault detection, intrusion detection, and so on. In our model, features were associated with external and immediately observable perceptual cues, but features may be computed based on other qualities, such as the behavior of nearby agents, proprioceptive sensory input, and environmental attributes. In this way, our CRM-based approach to classification in multiagent systems, could for instance give robots the capacity to distinguish between normal behavior and abnormal behavior. Since tolerance and its absence is determined online and does not require an initial training step, we expect that our CRM-based approach is particularly suitable to MRS operating in dynamic environments in which the task attributes may change over time and to MRS that adapt and change their behavior during task-execution. In this regard, we are currently investigating approaches to reduce the computational complexity of running the CRM on individual robots of a MRS.

Conclusions

In this study, we proposed an approach inspired by the capability of the adaptive immune system to maintain tolerance in multiagent systems. We further investigated the utility of this approach in task involving environment classification. Different environmental features were represented by different sensory stimuli in the environment, and their nature ("self" or "nonself") was not known by the agents beforehand i.e. it was not built into the individual agent's behavior. Our simulations revealed the capability of the collective of agents to tolerate features characterized as abundant and persistent, while mounting an immune response against specific features that were neither persistent nor abundant. Furthermore, the agent decision making was robust to perceptual noise and variations in their environmental conditions.

These encouraging results of our study provides a good stepping stone of our CRM-based approach for more detailed multiagent system experiments involving a broad

range of tasks.

Supplemental Data: Movies of MAS simulations are available online at <http://home.iscte-iul.pt/~alcen/alife2012/>.

Acknowledgment: This study was supported by the FCT grant PTDC/EEACRO/104658/2008.

References

- Abi-Haidar, A. and Rocha, L. (2008). Adaptive spam detection inspired by the immune system. In *Proceedings of the 11th International Conference on the Simulation and Synthesis of Living Systems, Artificial Life XI*, pages 1–8. MIT Press, Cambridge.
- Abi-Haidar, A. and Rocha, L. (2010). Biomedical article classification using an agent-based model of T-cell cross-regulation. In *Proceedings of the 9th International Conference on Artificial Immune Systems (ICARIS)*, pages 237–249. Springer-Verlag, Berlin, Germany.
- Abi-Haidar, A. and Rocha, L. (2011). Collective classification of textual documents by guided self-organization in T-cell cross-regulation dynamics. *Evolutionary Intelligence*, 4(2):69–80.
- Bradley, D. and Tyrrell, A. (2000). Hardware fault tolerance: an immunological solution. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, volume 1, pages 107 – 112. IEEE Press, Piscataway, NJ.
- Bradley, D. W. and Tyrrell, A. M. (2001). Immunotronics: Hardware fault tolerance inspired by the immune system. In *Proceedings of the 3rd International Conference on Evolvable Systems: From Biology to Hardware (ICES)*, pages 11–20. Springer-Verlag, Berlin, Germany.
- Butcher, J. (2003). *Numerical methods for ordinary differential equations*, chapter 23. John Wiley & Sons, West Sussex, England, second edition.
- Canham, R. and Tyrrell, A. (2002). A multilayered immune system for hardware fault tolerance within an embryonic array. In *Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS)*, pages 3 – 11. University of Kent at Canterbury Printing Unit.
- Carneiro, J., Leon, K., Caramalho, I., Van Den Dool, C., Gardner, R., Oliveira, V., Bergman, M., Sepúlveda, N., Paixão, T., Faro, J., and Demengeot, J. (2007). When three is not a crowd: a crossregulation model of the dynamics and repertoire selection of regulatory CD4⁺ T cells. *Immunological Reviews*, 216(1):48–68.
- Crespi, V., Galstyan, A., and Lerman, K. (2008). Top-down vs bottom-up methodologies in multi-agent system design. *Autonomous Robots*, 24(3):303–313.
- Evans, M., Hastings, N., and Peacock, B. (2000). *Statistical Distributions*, chapter 27, pages 134–136. John Wiley & Sons, New York, third edition.
- Hamann, H. (2010). *Space-Time Continuous Models of Swarm Robotic Systems: Supporting Global-to-Local Programming*. Cognitive Systems Monographs. Springer-Verlag, Berlin, Germany.

- Hauert, S., Zufferey, J., and Floreano, D. (2009). Evolved swarming without positioning information: an application in aerial communication relay. *Autonomous Robots*, 26(1):21–32.
- Janeway, C., Travers, P., Walport, M., and Shlomchik, M. (1997). *Immunobiology: The Immune System in Health and Disease*. New York: Garland Science.
- Kim, J. and Bentley, P. (1999). The human immune system and network intrusion detection. In *Proceedings of the 7th European Conference on Intelligent Techniques and Soft Computing (EUFIT)*. ELITE-Foundation, Aachen, Germany.
- Leon, K., Farob, J., and Carneiro, J. (2004). A general mathematical framework to model generation structure in a population of asynchronously dividing cells. *Journal of Theoretical Biology*, 229:455–476.
- Leon, K., Lage, A., and Carneiro, J. (2003). Tolerance and immunity in a mathematical model of t-cell mediated suppression. *Journal of Theoretical Biology*, 225:107–126.
- Leon, K., Perez, P., Lage, A., Farob, J., and Carneiro, J. (2000). Modelling t-cell-mediated suppression dependent on interactions in multicellular conjugates. *Journal of Theoretical Biology*, 207(2):231 – 254.
- Mermoud, G., Upadhyay, U., Evans, W., and Martinoli, A. (2010). Top-Down vs Bottom-Up Model-Based Methodologies for Distributed Control: A Comparative Experimental Study. In *Proceedings of the 12th International Symposium on Experimental Robotics (ISER)*, Springer Tracts in Advanced Robotics. Springer-Verlag, Berlin, Germany.
- Nino, F. and Beltran, O. (2002). A change detection software agent based on immune mixed selection. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC)*, pages 693–698. IEEE Computer Society, Washington, DC.
- O’Grady, R., Groß, R., Christensen, A. L., and Dorigo, M. (2010). Self-assembly strategies in a group of autonomous mobile robots. *Autonomous Robots*, 28:439–455.
- Parker, C., Zhang, H., and Kube, C. (2003). Blind bulldozing: multiple robot nest construction. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 2010–2015. IEEE Press, Piscataway, NJ.
- Parker, L. (2000). Current state of the art in distributed autonomous mobile robotics. In *Proceedings of the 5th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, pages 3–12. Springer, Berlin, Germany.
- Payton, D., Estkowski, R., and Howard, M. (2003). Compound behaviors in pheromone robotics. *Robotics and Autonomous Systems*, 44(34):229 – 240.
- Sahin, E. (2005). Swarm Robotics: From Sources of Inspiration to Domains of Application. In *Swarm Robotics*, volume 3342, pages 10–20. Springer, Berlin, Germany.
- Sakaguchi, S. (2004). Naturally arising CD4⁺ regulatory T cells for immunologic self-tolerance and negative control of immune responses. *Annual Review of Immunology*, 22:531–562.
- Tarapore, D., Floreano, D., and Keller, L. (2010). Task-dependent influence of genetic architecture and mating frequency on division of labour in social insect societies. *Behavioral Ecology and Sociobiology*, 64(4):675–684.
- Waibel, M., Keller, L., and Floreano, D. (2009). Genetic Team Composition and Level of Selection in the Evolution of Cooperation. *IEEE Transactions on Evolutionary Computation*, 13(3):648–660.
- Yamins, D. and Nagpal, R. (2008). Automated global-to-local programming in 1-D spatial multi-agent systems. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 615–622. IFAAMAS, Toronto, Canada.

Appendix A: Equations for activated T-cells

The section details the equations to calculate density of activated effector E_i^* and regulatory R_i^* cells, for all T-cell clones i . Given the conjugate density $C_{ij}(t)$ at steady state, the density of conjugated effector and regulatory cells is calculated proportional to the relative frequency of T_E and T_R cells in the clone. For the conjugated effector Ec_{ij} and regulatory Rc_{ij} cells of clone i at APC population j , we have:

$$Ec_{ij}(t) = \frac{C_{ij}(t)E_i(t)}{T_i(t)} \quad \text{and} \quad Rc_{ij}(t) = \frac{C_{ij}(t)R_i(t)}{T_i(t)}$$

Finally, for the number of activated effector E_i^* and regulatory R_i^* cells, we have:

$$E_i^* = \sum_{j=1}^M P_e(A_j, Ec_i, Rc_i) Ec_{ij} \quad (4)$$

$$R_i^* = \sum_{j=1}^M P_r(A_j, Ec_i, Rc_i) Rc_{ij} \quad (5)$$

where function P_e is the probability that an effector cell is conjugated with no neighboring regulatory cell at the same APC. P_r is the probability that a regulatory cell is conjugated with an APC that has at least one effector cell conjugated simultaneously. Additionally, Ec_i and Rc_i are the total number of conjugated effector and regulatory cells of clone i :

$$Ec_i = \sum_{j=1}^M Ec_{ij} \quad \text{and} \quad Rc_i = \sum_{j=1}^M Rc_{ij}$$

The probability functions P_e and P_r can be reduced to the following expressions, based on a multinomial approximation (Evans et al., 2000) that is valid given that the total number of sites (summed over all the APCs) is much larger than the number of sites per APC. For 3 binding sites ($s = 3$) on each APC, we have:

$$P_e(A_j, Ec_i, Rc_i) = \frac{(Rc_i - 3A_j)^2}{9A_j^2} \quad (6)$$

$$P_r(A_j, Ec_i, Rc_i) = \frac{(6A_j - Ec_i)Ec_i}{9A_j^2} \quad (7)$$

Utilizing the probability functions P_e and P_r , the density of effector and regulator cells can be calculated (eq 4 and 5).

Evolutionary Transitions and Top-Down Causation

Sara Imari Walker^{1,2,3}, Luis Cisneros⁴ and Paul C.W. Davies^{2,4}

¹NASA Astrobiology Institute, USA

² BEYOND: Center for Fundamental Concepts in Science, Arizona State University, Tempe AZ USA

³ Blue Marble Space Institute of Science, Seattle WA USA

⁴ Center for the Convergence of Physical Science and Cancer Biology, Arizona State University, Tempe AZ USA
sara.i.walker@asu.edu

Abstract

Top-down causation has been suggested to occur at all scales of biological organization as a mechanism for explaining the hierarchy of structure and causation in living systems (Campbell, 1974; Auletta et al., 2008; Davies, 2006b, 2012; Ellis, 2012). Here we propose that a transition from bottom-up to top-down causation – mediated by a reversal in the flow of information from lower to higher levels of organization, to that from higher to lower levels of organization – is a driving force for most major evolutionary transitions. We suggest that many major evolutionary transitions might therefore be marked by a transition in causal structure. We use logistic growth as a toy model for demonstrating how such a transition can drive the emergence of collective behavior in replicative systems. We then outline how this scenario may have played out in those major evolutionary transitions in which new, higher levels of organization emerged, and propose possible methods via which our hypothesis might be tested.

Introduction

The major evolutionary transitions in the history of life on Earth include the transition from non-coded to coded information (the origin of the genetic code), the transition from prokaryotes to eukaryotes, the transition from protists to multicellular organisms, and the transition from primate groups to linguistic communities (Szathmáry and Maynard Smith, 1997; Jablonka and Lamb, 2006). A hallmark of many of these transitions is that entities which had been capable of independent replication prior to the transition can subsequently only replicate as part of a larger reproductive whole (Szathmáry and Maynard Smith, 1995). A classic example is the origin of membrane bound organelles within modern eukaryotes, such as the mitochondria, which are believed to have emerged through endosymbiosis with prokaryotes that later lost their autonomy (Sagan, 1967). Each such transition is typically viewed as marking a drastic jump in complexity: cells are much more complex than any of their individual constituents (*i.e.* genes or proteins), eukaryotes are more complex than prokaryotes, multicellular more complex than unicellular organisms, and human societies more complex than individuals. However, although such a hierarchy is conceptually easy to state, in practice it

is difficult to determine what, if any, universal principles underlie such large jumps in biological complexity.

Szathmáry and Maynard Smith have suggested that all major evolutionary transitions involve changes in the way information is stored and transmitted (Szathmáry and Maynard Smith, 1995). An example is the origin of epigenetic regulation, whereby heritable states of gene activation lead to a potentially exponential increase in the amount of information that may be transmitted from generation to generation (since a set of N genes, existing in two states - on or off - via epigenetic rearrangements, can have 2^N distinct states). Such a vast jump in the potential information content of single cells is believed to have led to a dramatic selective advantage in unicellular populations capable of epigenetic regulation and inheritance (Jablonka and Lamb, 1995; Lachmann and Jablonka, 1996). The reasoning is straightforward: epigenetic factors permit a single cell line with a given genotype to express many different phenotypes on which natural selection might act, thereby providing a competitive advantage through diversification. Importantly, this innovation was crucial to the later emergence of multicellularity by permitting differentiation of many cell types from a single genomic inventory. However, although epigenetic regulation was likely a necessary precondition for the emergence of multicellular organization (at least in extant lineages), it does not necessitate that such a transition from unicellularity to multicellularity will occur or explain *how* it occurs. Plenty of protists are capable of phenotypic differentiation but have never made the transition to true multicellularity, although they may exhibit highly collective and coordinated behaviors (see *e.g.* Nedelcu and Michod (2004) for a discussion of unicellular, multicellular, and a gamut of intermediate forms within the Volvocacean green algal group). More generally, while it is true that changes in how information is stored and transmitted enable the possibility of new levels of organization to evolve, such innovations are not necessarily a sufficient causative agent to drive the emergence of genuinely new, higher-level, entities.

Therefore to make progress in understanding the major transitions, a key, and oft understated, distinction must be

made between the evolutionary innovations leading up to a major transition that enable higher-levels of organization to emerge, and the mechanism(s) underlying the physical transition itself. In general, the majority of unifying work on major evolutionary transitions has focused on the former perspective by outlining the key steps that enabled a particular transition to occur, such as the innovations in information storage and transmission outlined by Szathmáry and Maynard Smith (Szathmáry and Maynard Smith, 1997). While these innovations are certainly crucial to our understanding of the historical sequence of evolutionary events surrounding each major transition – such as the example of the appearance of phenotypic differentiation via epigenetic regulation prior to the emergence of multicellularity cited above – they tell us little about the underlying mechanisms governing the emergence of genuinely new higher-levels of organization. If there are in fact any universal principles common to all such major jumps in biological complexity, we should expect there to be a common mechanism driving each such transition that is not dependent on a precise series of historical (evolutionary) events. In this paper, we focus on those major evolutionary transitions leading to the emergence of new, higher level entities, which are composed of units that previously reproduced autonomously. We propose that these major transitions, corresponding to major jumps in biological complexity, are associated with *information gaining causal efficacy over higher levels of organization*.

To outline our hypothesis, we first present an introduction to top-down causation in biological systems, and outline how a transition to top-down causation via informational efficacy over new, higher levels of organization might enable the emergence of higher-level entities. We then present a toy model, investigating the onset of non-trivial collective behavior in a globally coupled logistic map lattice, to demonstrate how a reversal in the dominant direction of information flow, from bottom-up to top-down, is correlated with the emergence of collective behavior in replicative systems. A key feature of our analysis is to determine the direction of causal information transfer. We then outline how a transition in causal structure may have driven both the origin of life and the origin of multicellularity, as two representative examples of major evolutionary transitions in which new, higher levels of organization emerged. We conclude with some suggestions about possible methods via which our hypothesis might be tested.

Informational Efficacy and Top-Down Causation

Biological information is a notoriously difficult concept to define (Küppers, 1990). This difficulty stems in part from the fact that in living systems the dynamics are coupled to the information content of biological states such that the dynamics of the system change with the states and vice versa (Goldenfeld and Woese, 2011; Davies, 2012). This is in

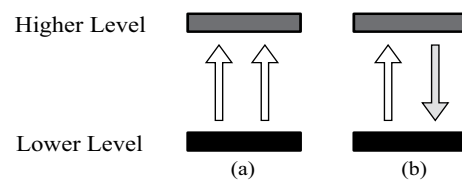


Figure 1: Bottom-up and top-down modes of causation. (a) The standard (reductionist) view suggests everything in the universe is directed by bottom-up action only, such that causation flows strictly from lower to higher levels. (b) Biological organization suggests an alternative causal structure whereby bottom-up modes of causation emanating from lower-levels provide a space of possibilities, and higher-levels of organization modify the causal relations below via top-down causation. Figure adapted from (Auletta et al., 2008).

marked contrast to the traditional approach to dynamics, where the physical states evolve with time but the dynamical laws remain fixed, or change over much longer time scales. The coupling of states to dynamics is perhaps most evident for the case of the genome, in which the expressed set of instructions – *i.e.* the relative level of gene expression – depends on the state of the system – *i.e.* the composition of the proteome, environmental factors, *etc.* – that regulate the switching on and off of individual genes. The result is that the update rules change with time in a manner which is both a function of the current state and the history of the organism (Goldenfeld and Woese, 2011). This feature of “dynamical laws changing with states” (Davies, 2012), as far as we know, seems to be unique to biological organization and is a direct result of the peculiar nature of biological information (although speculative examples from cosmology have also been discussed, see *e.g.* (Davies, 2006a)).

Biological information is distinctive in its contextual or semantic nature, in other words it *means* something (Maynard Smith, 2000). For example, a gene is just a random sequence of nucleotides when taken in isolation, and is indistinguishable from junk, or noncoding, DNA. It is meaningful, or biologically functional, only within the context of the cell, where a suite of molecular hardware collaborate in decoding and executing the encoded instruction (*e.g.* to make a protein). As such, biological information is an abstract global systemic entity, carrying meaning only within the context of an entire living system. It is of course imprinted in biochemical structures, but one cannot point to any specific structure in isolation and say “aha! I see biological information here!”: even the information in genes is only efficacious and manifested in a relational sense (*i.e.* it must be decoded by the appropriate cellular machinery). Perhaps even more profound, this abstraction appears to have causal efficacy (Auletta et al., 2008; Ellis, 2012; Davies, 2012) – it

is the information that determines the state and hence the dynamics. As such, it is the efficacy of information that leads to the convolution of dynamical laws and states that makes biology so unique.

This convolution results in multidirectional causality with causal influences running both up and down the hierarchy of structure of biological systems (*e.g.* both from genome to proteome, *and* from proteome to genome via the switching on and off of genes). A full explanatory framework for biological processes should therefore include both bottom-up causation (Fig. 1a) – such as when a gene is read-out to make a protein that affects cellular behavior – and top-down causation (Fig. 1b) – as occurs when changes in the environment initiate an organismal response that permeates all the way down to the level of individual genes (Davies, 2012). A striking of the latter is provided by the phenomenon of “mechanotransduction”, where physical forces, such as the sheer stress on a cell or the Youngs modulus of an adjacent surface to which the cell attaches, actually affect gene expression (Alberts et al., 2002). Bottom-up causation is the status-quo in modern physics, whereas top-down causation is less familiar and difficult to quantify. Generally, top-down causation is characterized by a ‘higher’ level influencing a ‘lower’ level by setting a context (for example, by changing some physical constraints) by which the lower level actions take place (Auletta et al., 2008; Davies, 2006b; Ellis, 2006, 2012). An interesting example of top-down causation is provided by natural selection in evolution (Campbell, 1974; Okasha, 2012), where the history as well as the fate of an organism is determined by the wider environmental context. This is particularly evident for cases of convergent evolution (Davies, 2006b), of which the wing is a classic example. Birds, pterodactyls, and bats each developed wings, despite the fact that their last common ancestor did not possess wings. The commonality of form is attributable to physical (environmental) constraints imposed on wing design, which manifests a particular phenotypic trait in the organism (*i.e.* a wing). However, the effect is also a local physical one: the biochemical interactions – dictated in part by both genetic and epigenetic programming – that govern the morphological development of something as complex as a bird wing are inherently local. As such, natural selection provides a well-known example of how higher level processes (*e.g.* environmental selection) constrain and influence what happens at lower levels (*e.g.* biochemistry).¹

¹Although it is normal for biologists to discuss causal narratives in informational terms (*e.g.* cells signal each other, and recruit molecules to express instructions ...) a determined reductionist would argue that, in principle, this narrative would parallel an, albeit vastly more complicated, account in terms of molecular interactions alone, in which only material objects enjoy true causal efficacy. In this paper we remain agnostic on the question of such promissory reduction because our principal claims remain valid even if the informational-causal narrative is accepted as a mere *façon de parler*.

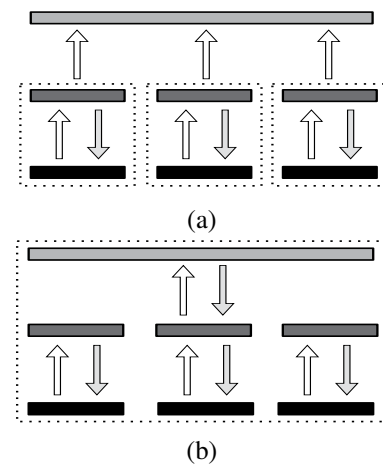


Figure 2: Schematic illustrating a shift in causal structure mediated by the transition from a collective of lower level entities to a new higher level entity. (a) Prior to the transition, higher levels of organization are dictated by bottom-up causation directed by lower level entities (which themselves may be hierarchal in nature). (b) A new (higher level) entity emerges that may be identified as an “individual” when a transition to top-down causation with efficacy over a new, higher-level of organization (in this case the medium grey level) occurs. Dotted lines are used to indicate an individual entity (*i.e.* an organism as in the case for the transition from unicellular to multicellular organisms)

The foregoing discussion indicates that top-down causation – mediated by informational efficacy – plays an important role in dictating the dynamics of living systems, where causal influences can run both up and down structural hierarchies. In many of the major evolutionary transitions, new higher-level entities emerge from the collective and coordinated behavior of lower-level entities, eventually transitioning to a state of organization where the lower-level entities no longer have reproductive autonomy. Examples of transitions where lower-level units have lost their autonomy include the the origin of life (*i.e.* the hypothetical RNA world), the origin of multicellularity, and the origin of eusociality (Szathmáry and Maynard Smith, 1997). During such transitions, the dynamics of lower level entities come under the direction of the emergent high-level entity. This, coupled with the multidirectional causal influences in biological organization, suggests that evolutionary transitions that incorporate new, higher levels of organization into a biological system *should be characterized by a transition from bottom-up to top-down causation, mediated by a reversal in the dominant direction of information flow*. Therefore, we suggest that major shifts in biological complexity – from lower level entities to the emergence of new, higher level entities – are associated with a physical transition (perhaps akin to a ther-

modynamic phase transition), and this physical transition is in turn associated with a fundamental change in causal structure (schematically illustrated in Fig. 2). To illustrate this claim, we turn to a toy model investigating the emergence of non-trivial collective behavior in a globally coupled chaotic map lattice (Cisneros et al., 2002; Ho and Shin, 2003).

Logistic Growth as a Toy Model

To explore the emergence of non-trivial collective behavior and its connection to transitions in causal structure, we focus on a lattice of chaotic logistic maps. The logistic growth model was chosen for its connection to the replicative growth of biological populations (Murray, 1989), thereby enabling us to make an analogy with the transition from independent replicators to collective reproducers, cited as a hallmark of many major evolutionary transitions as outlined above (Szathmáry and Maynard Smith, 1995). Our aim with this simplified model is to provide a clear example of how a reversal in information flow – from bottom-up to top-down – can describe a transition from a group of independent low-level entities to the emergence of a new higher-level (collective) entity.

Our model system is defined as

$$x_{i,n+1} = (1 - \epsilon)f_i(x_{i,n}) + \epsilon m_n \quad ; \quad (i = 1, 2, \dots, N) \quad (1)$$

where the function $f_i(x_{i,n})$ specifies the local dynamics of element i , N is the total number of elements, n is the current time-step (generation), and ϵ is the global coupling strength to the instantaneous dynamics of the mean-field, m_n , defined below in eq. 4. In analogy with biological populations, the element index i may be associated with a specific phenotype within a given population, and ϵ marks the strength of the global informational control over the local dynamics of each such element. The local dynamics of each element i is defined by the discrete logistic growth law

$$f_i(x_{i,n}) = r_i x_{i,n} \left(1 - \frac{x_{i,n}}{K}\right) \quad (2)$$

where r_i is the reproductive fitness of population i , and K is the carrying capacity – set to $K = 100$ for all i , for the results presented here. The instantaneous state of the entire system at time step n is specified as an average over all local states by the instantaneous mean-field M_n ,

$$M_n = \frac{1}{N} \sum_{j=1}^N x_{j,n} \quad (3)$$

and the instantaneous dynamics of the mean-field,

$$m_n = \frac{1}{N} \sum_{j=1}^N f_j(x_{j,n}) \quad (4)$$

is a global systemic entity (*i.e.* it cannot be identified with any specific local attribute), which has direct impact on the

dynamics of local elements i in our model system. The influence of this abstract global entity is dictated by the global coupling strength ϵ .

The system was initialized with $x_{i,0} = 1$ for all elements i , representing an initial population size of one individual for each population. Values for the fitness parameters r_i were randomly drawn from the range of values $[3.9, 4.0]$, where selection of replicative fitness was restricted to this range to ensure that all elements individually display chaotic dynamics even when coupled to the global dynamics (required to determine cause and effect for this model system, see *e.g.* Cisneros et al. (2002)). Following the dynamics of a set of $N = 1000$ coupled logistic maps, a time series of both the instantaneous states of the local elements, $x_{i,n}$, and of the mean-field, M_n , was generated from which causal directionality and the associated flow of information were determined. In what follows, we introduce a definition of a measure of causal information transfer based on analyses of multivariable time series and then present results for the causal structure of our coupled logistic growth model using this measure.

Quantifying Causal Information Transfer

Standard measures of information, such as Shannon entropy (Shannon and Weaver, 1949), which provides the average number of bits needed to encode independent events of a discrete process, and mutual information, used to measure the joint probability of two process, rely on static probabilities. However, in order to infer causal information transfer (*i.e.* from higher to lower levels versus from lower to higher levels of organization, or here from the mean-field to local elements versus from local elements to the mean-field), a measure that can capture *dynamical structure* by means of transition probabilities rather than static probabilities is required. The dynamical character of the interactions can be studied by introducing a time lag in order to compute the relevant transition probabilities.

Consider, for example, a Markov process of order k . The conditional probability $p(x_{n+1}|x_n, \dots, x_{n-k+1}) = p(x_{n+1}|x_n, \dots, x_{n-k+1}, x_{n-k})$ describes a transition probability whereby each state x_{n+1} of the process is dependent (conditional) on the last k -states but is independent of the state x_{n-k} and all previous states. This conditional relationship can be extended to any k -dimensional dynamical system as prescribed by Takens embedding theorem (Takens, 1980). To simplify notation, we define an embedded state as $x_n^{(k)} = (x_n, \dots, x_{n-k+1})$, which describes a state in the k -dimensional phase space, such that the series of vectors $\{x_n^{(k)}\}$ contains all of the information necessary to characterize the trajectory of the dynamical variable x . Using this definition, the dynamical information shared between two processes, x and y , can be determined by the Transfer En-

tropy (Schreiber, 2007):

$$T_{Y \rightarrow X}^{(k)} = \sum_n p(x_{n+1}, x_n^{(k)}, y_n^{(k)}) \log \left[\frac{p(x_{n+1} | x_n^{(k)}, y_n^{(k)})}{p(x_{n+1} | x_n^{(k)})} \right] \quad (5)$$

This measure incorporates causal relationships by relating delayed (embedded) states, $x_n^{(k)}$ and $y_n^{(k)}$, to the state x_{n+1} , and quantifies the incorrectness of assuming independence between the two processes x and y . In short, the transfer entropy tells us the deviation from the expected entropy of two completely independent processes.

The transition probabilities can be systematically measured from the time series by coarse graining the phase space. Calculation of the conditional probabilities

$$p_r(x_{n+1} | x_n^{(k)}, y_n^{(k)}) = \frac{p_r(x_{n+1}, x_n^{(k)}, y_n^{(k)})}{p_r(x_n^{(k)}, y_n^{(k)})} \quad (6)$$

$$p_r(x_{n+1} | x_n^{(k)}) = \frac{p_r(x_{n+1}, x_n^{(k)})}{p_r(x_n^{(k)})} \quad (7)$$

then yields all of the necessary quantities required to calculate the transfer entropy $T_{Y \rightarrow X}^{(k)}$ as defined in equation (5). Larger values for the information transfer are expected to be measured when the defined embedded space is a better representation of the real phase space of the dynamical process that generates the set of states $\{x_n\}$. Therefore, selection of the dimension of embedding k is done such that $T_{Y \rightarrow X} = \text{Max}\{T_{Y \rightarrow X}^{(k)}\}$.

Information Flow Between Global and Local Scales

The coupled system described by eqs. (1), (2), and (4) displays several different phases with interesting dynamical properties (Fig. 3). A detailed description of the dynamical features of these phases is outlined in the study by Balmforth *et al.* (Balmforth et al., 1999). Here we focus our discussion on the observed collective behavior in the context of the measured causal information transfer characterized in eq. (5). We compare the flow of information from local to global scales and from global to local scales – $T_{X \rightarrow M}$ and $T_{M \rightarrow X}$ respectively – to demonstrate how causal information transfer from the global to the local dynamics corresponds to the emergence of collective organization.

The time series for 1,000 logistic maps were recorded for ten thousand generations (time-steps), including the time series of the mean field, M , and that of an arbitrarily chosen local element, x , selected at random to be representative of typical local dynamical behavior. The dynamics of the system varies widely as a function of the coupling parameter ϵ , indicative of variations in the degree to which the local and mean-field dynamics influence the dynamics of individual local elements. For $\epsilon = 0$, the system is completely uncoupled (each local element acts independently), and the dynamics are that of 1,000 isolated subpopulations. The

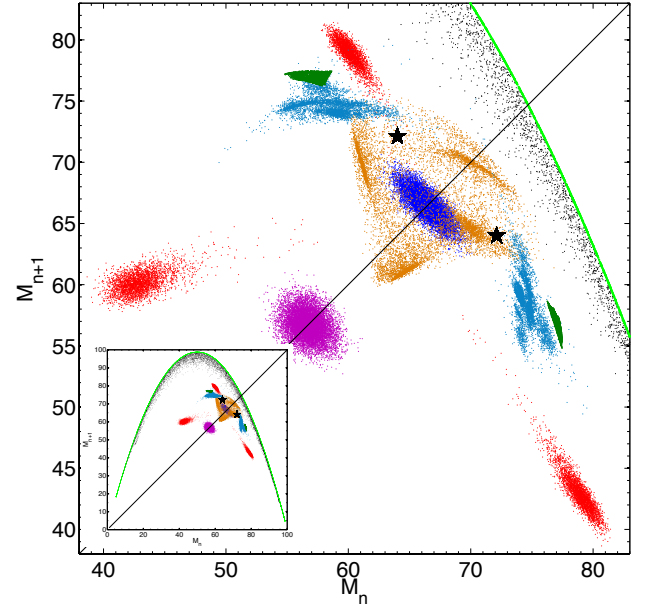


Figure 3: Return map for varying values of the global coupling strength ϵ . Shown are return maps for $\epsilon = 0$ (magenta), $\epsilon = 0.075$ (red), $\epsilon = 0.1$ (blue), $\epsilon = 0.2$ (orange), $\epsilon = 0.225$ (aqua), $\epsilon = 0.25$ (dark green), $\epsilon = 0.3$ (stars), and $\epsilon = 0.4$ (black). Also shown is the return map for a single logistic map (bright green). The inset shows an expanded view.

opposite extreme, $\epsilon = 1$, corresponds to complete coupling, where all the logistic maps evolve identically to each other with fully synchronized dynamics (*i.e.* they may be identified as part the same higher-level “organisms”).

As the coupling strength is increased from no coupling at $\epsilon = 0$, a rich diversity of self-organized collective phenomena are observed to emerge. A sampling of this variety are detailed by the return-map of the mean field M , shown in Fig. 3. For $\epsilon = 0$ (the uncoupled limit), the return-map of the mean-field is a cloud of dispersed points around a fixed value (Fig. 3, magenta), as is characteristic of dynamics with random oscillations about a fixed value. For $\epsilon = 0.075$ (Fig. 3, red) a clear quasi-periodic three state oscillatory dynamic is observed, as evidenced by the three clouds in the return map (with some dispersion), indicating that the system has achieved a moderate degree of collectivity. Although the coupling strength is relatively low, the system self-organizes in such a way that the mean field has a simpler dynamic than the typical chaotic behavior of the individuals. The system organizes by forming clusters, within which the individuals have very similar behavior. Here it is likely that top-down information transfer is highest within clusters, resulting in intermediate size scales (between local and global) driving the emergence of collective behavior: this dynamic is not accurately captured by our global measure M . As such, the transfer entropy, shown in Fig. 4 for top-down ($T_{M \rightarrow X}$,

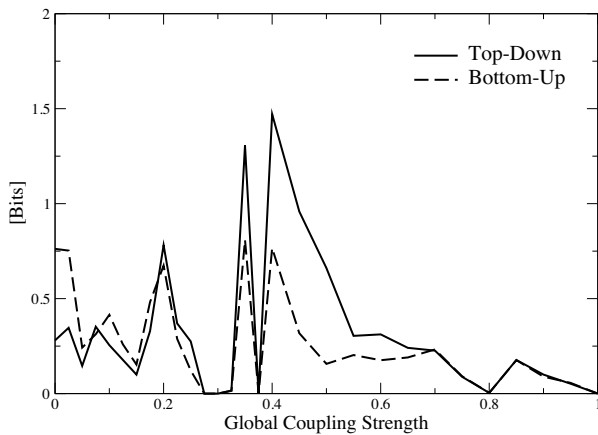


Figure 4: Top-down, $T_{M \rightarrow X}$ (solid), and bottom-up $T_{X \rightarrow M}$ (dashed), causal information transfer for varying global coupling strength ϵ of a system of coupled logistic maps.

solid) and bottom-up ($T_{X \rightarrow M}$, dashed), do not quite reflect the onset of this collective dynamic, although $T_{X \rightarrow M}$ and $T_{M \rightarrow X}$ are nearly equal, indicating that the dynamics of the global mean-field is driving, at least partially, the collective behavior in a top-down manner.

Increasing the coupling strength to $\epsilon = 0.1$, the system falls back into the dynamics of a seemingly disorganized system, and no particularly interesting global behavior is observed. This is shown in the return map as a randomly dispersed cloud around a fixed value (Fig. 3, blue). However, further increasing the coupling strength to $\epsilon = 0.2$, yields the onset of a new collective phase (Fig. 3, orange). Here, collective behavior manifests as four phase periodic oscillation with large dispersion. A small dominance of the top-down transfer entropy $T_{M \rightarrow X}$ is observed. Here, the onset of collective behavior corresponds to a transition to top-down information flow being the dominate mechanism of information transfer. This provides the first presented example of a clear case where top-down causation drives the emergence of collective behavior. It is interesting that this occurs at a fairly low value of the coupling strength, at $\epsilon = 0.2$.

The attractor observed for $\epsilon = 0.2$ breaks apart into two dispersed clouds for $\epsilon = 0.225$ (Fig. 3, aqua) and then concentrates into smaller clouds for $\epsilon = 0.25$ (Fig. 3, dark green), where the system enters a collective two-phase periodic oscillation. These observed collective phases also correspond to the top-down transfer entropy being the dominant causal driving force, as shown in Fig. 4. Increasing ϵ to $\epsilon = 0.3$ leads to complete synchronization of the system (Fig. 3, stars), yielding a transfer entropy measure of zero. In general, we expect both $T_{M \rightarrow X}$ and $T_{X \rightarrow M}$ to be zero for states of complete synchronization since no information can be gained from a coding by considering a second time

series that is dynamically identical to the first one. In other words, when full synchronization is achieved, two series become dynamically identical and it no longer makes sense to discuss transfer entropy between them. In this particular case, full synchronization indicates the local dynamics are coincident with the mean-field, *i.e.* a transition to a fully collective emergent entity has occurred. It is interesting that these dynamics are first observed for such a low value of ϵ , and that increasing ϵ still further yields states which are not fully synchronized. Additionally, in this regime where synchronization emerges dynamically, any collective dynamics in which transfer entropy can be measured (*i.e.* not synchronized) are dominated by top-down transfer entropy (*e.g.* see $\epsilon = 0.25$ and 0.35 in Figure 4), suggesting that the dynamical synchronization occurs due to top-down dynamical driving. For $\epsilon = 0.4$, the return map approaches the form of the chaotic attractor for a logistic map (Fig 3, black), indicating that the dynamics are collectively logistic. The mean field is dynamically chaotic, as are the individual maps in the lattice; however, here the individual maps are not synchronized with each other, yielding non-trivial organization. The emergence of this highly collective state is again reflected by a dominance of the top-down transfer entropy relative to the bottom-up transfer entropy. This trend is continued for increasing ϵ until $\epsilon = 0.7$, at which point the mean-field and local dynamics are fully synchronized and it no longer makes sense to discuss information transfer, as noted above.

In general, the trends observed indicate that each time a collective state emerges, causal information transfer is dominated by information flow from global to local scales. Particularly interesting is that top-down causation dominates for collective states in regimes with $0.2 < \epsilon < 0.7$, where the contribution from the global dynamics is not necessarily the dominant contribution in eq. (1) (*i.e.* for $0.2 < \epsilon < 0.5$). In this regime, although the weight of the contribution from the global scale may be less than the contribution from the local scale in dictating the local dynamics, collective states self-organize which are driven by top-down causal information transfer from the mean-field. Although we have focused on a coupling to the global mean-field for the work presented here, other studies of coupled chaotic map lattices have shown that strictly local coupling leads to similar dynamical behavior (Cisneros et al., 2002; Ho and Shin, 2003) - *i.e.* even in cases where the mean-field never appears in the dynamical equations, the global dynamics can still drive the emergence of collective behavior via top-down causation.

Major Transitions in Causal Structure

The results presented for this toy model system indicate that a transition from a population of independent replicators, to a collective representing a higher-level of organization, can be mediated by a physical transition from bottom-up to top-down information flow, where non-trivial collective behavior is associated with the degree to which local ele-

ments receive information from the global network. The dynamical system investigated was designed to parallel transitory dynamics believed to be a hallmark feature of many major evolutionary transitions – *i.e.* those characterized by the emergence of higher-level reproducers from lower level units (Szathmáry and Maynard Smith, 1995). For the model system presented above, new high-level entities would be expected to emerge as $\epsilon \rightarrow 1$ (although non-trivial collective behavior is observed to emerge in intermediate regimes, as discussed above). Examples of major evolutionary transitions where similar dynamics are expected to have played out include the origin of life, the origin of eukaryotes, the origin of multicellularity, and the origin of eusociality. Here we focus on discussing the origin of life and the origin of multicellularity as two representative examples of major evolutionary transitions that may potentially be driven by transitions in causal structure as dictated by informational gaining efficacy over higher-levels of organization.

The Origin of Life. In the original classification scheme of Szathmáry and Maynard Smith, three major transitions are associated with the origin of life: from replicating molecules to populations of molecules in compartments, from unlinked replicators to chromosomes, and from RNA to RNA + DNA + protein (*i.e.* the origin of the genetic code). However, given that we do not know the specific sequence of events leading to the emergence of the first known life, a more pragmatic perspective is to assume that when life as we know it first emerged, it was surely characterized by the same distinctive hierarchical and causal structure as all known life. Adopting this viewpoint, Walker and Davies have recently suggested that a transition in causal structure, from bottom-up to top-down, was the critical step in the origin of life (Walker and Davies, 2012). In this context, the origin of life is associated with the emergence of a collective contextual information processing system with top-down causal efficacy over the matter it is instantiated in (Walker and Davies, 2012). The transition from non-living to living matter may therefore be identified when information (stored in the state of the system) gains causally efficacy. A constructive measure of how close chemical systems are to the living state – a quantity notoriously absent in almost all discussions of the origin of life – may therefore be provided by adopting a variant of the parameter ϵ and applying it to the relevant chemical kinetics. This may provide new avenues of research into the origin of life by directing efforts toward understanding how chemical systems come under direction of the global context rather than focusing strictly on the evolutionary processes that might enable a transition to the living state but do not necessitate it.

The Origin of Multicellularity. Unlike the emergence of life, where the frequency of origination events is entirely unknown, multicellularity is believed to have arisen dozens of

times in the history of life on Earth (Bonner, 1999). A possible explanation for the numerous transitions to multicellularity is that many of the hallmarks of multicellular organisms are laid out by epigenetic factors and physical effects in unicellular aggregates that only later come under information (*i.e.* genetic) control. For example, Newman and collaborators have proposed that the variety of metazoan body plans were originally laid out by physical interactions, such that the phenotype of multi-cellular aggregates was determined at first by physical environmental influences (Newman and Müller, 2000; Newman et al., 2006). They suggest that these physical varieties of form were only later to be taken over by innovations in genetic programming. An explicit example of a similar process whereby information control dictates the emergence of collective states is provided within the genus *Volvox*: the multicellular green alga, *Volvox carteri* has a gene controlling cellular differentiation that is related to an analogous gene dictating cellular phenotype in its unicellular relative, *Chlamydomonas reinhardtii* (Nedelcu, 2009), which may have played a crucial role in its transition to multicellularity. This suggests that a key feature of the transition to multicellular organization is biological information gaining efficacy over new scales of organization by redirecting features already present in collectives of the lower-level units. As such, the physical transition should be marked by a transition in causal structure. An interesting consideration is therefore that multicellularity emerges frequently, requiring only the physical transition from bottom-up to top-down causation via information control once the underlying lower-level units possess evolutionary innovations necessary to prime them for the transition. An important question is then: how hard is it for the physical transition to occur? From the viewpoint of the perspective provided here, further investigations into the causal structure of biological systems are required to address this question. The relevant order parameter ϵ could be measure of the degree of signaling between individual cells (*i.e.* their response to intercellular signaling), or a measure reproductive viability as presented with the simple logistic growth model detailed above. In general this approach requires innovations in understanding the degree to which the whole dictates the parts in biological collectives, as much as understanding the degree to which the parts dictate the whole.

Given that we do not have a clear picture of the causal structure of biological systems, it is at present unclear what the relative role of bottom-up and top-down causative effects are in directing biological organization. Here we have proposed that increasing levels of biological complexity, corresponding to increased depth in the hierarchical organization of living systems, correspond to information gaining causal efficacy over increasingly higher levels of organization. Each major evolutionary transition leading to the emergence of genuinely new, higher-level entities from lower-level units, should therefore be characterized by a transition

in causal structure mediated by a reversal in the dominant direction of information flow from bottom-up to top-down. We have demonstrated the dynamics of such a transition by appealing to a toy system of coupled logistic maps. The dynamics observed verify that collective states emerge in association with a transition to top-down causal information transfer as the dominant direction of information flow. The nature of the reversal in causal structure presented here suggests that biological systems cannot jump up the ladder of hierarchical structure - information must first gain control over a lower-level of organization before the emergence of efficacy over higher-levels can take-hold. Rapid diversification may occur after each such transition due to the new capacity for directing physical processes at the higher level.

Acknowledgements

SIW gratefully acknowledges support from the NASA Astrobiology Institute through the NASA Postdoctoral Fellowship Program. LC and PCWD were supported by NIH grant U54 CA143682.

References

- Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2002). *Molecular Biology of the Cell*, page 1357. Garland Science, New York, 4 edition.
- Auletta, G., Ellis, G. F. R., and Jaeger, L. (2008). Top-down causation by information control: from a philosophical problem to a scientific research programme. *J. R. Soc. Interface*, 5(27):1159–72.
- Balmforth, N. J., Jacobson, A., and Provenzale, A. (1999). Synchronized family dynamics in globally coupled maps. *Chaos*, 9:738–754.
- Bonner, J. T. (1999). The origins of multicellularity. *Integrative Biol.*, 1:27–36.
- Campbell, D. (1974). Downward causation in hierarchically organized biological systems. In Ayala, F. J. and Dobzhansky, T., editors, *Studies in the philosophy of biology: Reduction and related problems*, pages 179–186. Macmillan, London, UK.
- Cisneros, L., Jiménez, J., Cosenza, M. G., and Parravano, A. (2002). Information transfer and nontrivial collective behavior in chaotic coupled map networks. *Phys. Rev. E*, 65:045204.
- Davies, P. C. W. (2006a). *The Goldilocks Enigma*. Penguin Books, New York.
- Davies, P. C. W. (2006b). The physics of downward causation. In Clayton, P. and Davies, P. C. W., editors, *The re-emergence of emergence*, pages 35–52. Oxford University Press, Oxford, UK.
- Davies, P. C. W. (2012). The epigenome and top-down causation. *J. R. Soc. Interface*, 2(1):42–48.
- Ellis, G. F. R. (2006). On the nature of emergent reality. In Clayton, P. and Davies, P. C. W., editors, *The re-emergence of emergence*, pages 79–107. Oxford University Press, Oxford, UK.
- Ellis, G. F. R. (2012). Top-down causation and emergence: some comments on mechanisms. *J. R. Soc. Interface*, 2(1):126–140.
- Goldenfeld, N. and Woese, C. (2011). Life is Physics: Evolution as a Collective Phenomenon Far From Equilibrium. *Annual Review of Condensed Matter Physics*, 2(1):375–399.
- Ho, M.-C. and Shin, F.-C. (2003). Information flow and nontrivial collective behavior in chaotic-coupled-map lattices. *Phys. Rev. E*, 67:056214.
- Jablonka, E. and Lamb, M. J. (1995). *Epigenetic Inheritance and Evolution*. Oxford University Press, Oxford.
- Jablonka, E. and Lamb, M. J. (2006). The evolution of information in the major transitions. *J. Theor. Biol.*, 239(2):236–46.
- Küppers, B. (1990). *Information and the Origin of Life*. MIT Press, Cambridge.
- Lachmann, M. and Jablonka, E. (1996). The inheritance of phenotypes: an adaptation to fluctuating environments. *Journal of Theoretical Biology*, 181(1):1–9.
- Maynard Smith, J. (2000). The concept of information in biology. *Philosophy of Science*, 67(2):177–194.
- Murray, J. D. (1989). *Mathematical Biology*. Springer-Verlag, Berlin.
- Nedelcu, A. and Michod, R. E. (2004). Evolvability, modularity, and individuality during the transition to multicellularity in volvocalean green algae. In Schlosser, G. and Wagner, G., editors, *Modularity in Development and Evolution*, pages 466–489. University of Chicago Press, Chicago, IL.
- Nedelcu, A. M. (2009). Environmentally induced responses co-opted for reproductive altruism. *Biol. Lett.*, 5(6):805–808.
- Newman, S. A., Forgacs, G., and Muller, G. B. (2006). Before programs: the physical origination of multicellular forms. *Int. J. Dev. Biol.*, 50(2-3):289–99.
- Newman, S. a. and Müller, G. B. (2000). Epigenetic mechanisms of character origination. *J. Exp. Zool.*, 288(4):304–317.
- Okasha, S. (2012). Emergence, hierarchy and top-down causation in evolutionary biology. *J. R. Soc. Interface*, 2(1):49–54.
- Sagan, L. (1967). On the origin of mitosing cells. *J. Theor. Biol.*, 14(3):225.
- Schreiber, T. (2007). Measuring information transfer. *Phys. Rev. Lett.*, 85:461.
- Shannon, C. E. and Weaver, W. (1949). *The Mathematical Theory of Information*. University of Illinois Press, Urbana IL.
- Szathmáry, E. and Maynard Smith, J. (1995). The major evolutionary transitions. *Nature*, 374(6519):227–232.
- Szathmáry, E. and Maynard Smith, J. (1997). *The major transitions in evolution*. Oxford University Press, Oxford.
- Takens, F. (1980). Detecting strange attractors in turbulence. *Dynamical Systems and Turbulence*, 898:366.
- Walker, S. I. and Davies, P. C. W. (2012). The algorithmic origins of life. In prep.

The Role of Working Memory in an Urban Pursuit Scenario

Ransom K. Winder¹, and James A. Reggia²

¹The MITRE Corporation

²University of Maryland, College Park
rwinder@mitre.org

Abstract

Most self-organizing models of moving agent collectives (simulated herds, bird flocks, etc.) employ reflexive agents that lack significant memory of past movements and previously encountered environmental features. Further, these agent collectives often act in fairly open environments where obstacles to movement are relatively sparse. In this work, we explore the hypothesis that a limited working memory of recently encountered environmental features, distributed throughout the collective, can improve task performance for a team of interacting agents that are operating in a highly occluded environment. Investigating a team of agents pursuing a mobile target in an “urban environment”, we found that the team benefited from 1) communication that coordinated team movements, and 2) from a working memory of the environment that was distributed among the agents, despite individual agents knowing only a small part of the relevant information. These results further our understanding of methodologies that can be applied to control robotic teams and swarm optimization, and may also provide insight into herd behavior of biological populations in some densely occluded environments.

Introduction

Teams of collectively moving agents have been widely studied in artificial life for many years. Reynolds’ early work established that basic agent interactions, such as avoidance, alignment and cohesion forces, could produce surprisingly realistic flock-like behaviors (Reynolds, 1987). Subsequent studies have extended these results in areas such as robotics (Atherton, et al. 2006; Bayazit, et al. 2002; Couzin, et al. 2005; McCook and Esposito, 2007) and optimization (Kennedy et al, 2001; Lapiroz-Encinas, Kingsford, and Reggia, 2009), and have shown that they can be integrated effectively with goal-driven control mechanisms to support problem solving (Rodriguez and Reggia, 2005; Rodriguez and Reggia, 2009). Other formulations have been explored, such as the use of potential fields that guide agent movements (Vail and Veloso, 2003; Kurihara, et al. 2005).

While a great deal has been learned about collective movements from these past studies, most past multi-agent systems have involved movements that occur in largely open spaces in which objects that act as obstacles are relatively sparse. Much less is known about collective movements in densely occluded spaces, such as urban environments where an agent would be limited to moving between buildings and to having only very local visual information. It is not even clear

a priori whether or not collective movements in the latter situation are advantageous relative to independent agent movements. Further, while maintaining a partial memory of past obstacles can be useful to agent teams in relatively open settings (Winder and Reggia, 2004), it is not clear whether or not this remains the case when movements are highly constrained by ubiquitous environmental barriers.

To help clarify these issues, here we consider a multi-agent pursuit task in a highly constrained “urban environment”, or simulated city. This pursuit scenario requires multiple agents to work together to capture a moving target with capabilities on par with those of the pursuing agent team. At issue is whether or not capturing the target can be facilitated through communication between agents, coordination of their movement behaviors, and experiential knowledge (working memory) of the environment. For single agents, an episodic memory can provide a benefit (Nuxoll, 2011), but our goal here is to assess whether there is a benefit from a more volatile working memory. The urban pursuit scenario we use can be related in spirit to other past predator-prey systems (Benda, et al. 1986; Alcazar, 2004; Lenzitti, et al. 2005; Zhao and Jin, 2005; Hladek, et al. 2009; Huang, et al. 2009), but is specifically oriented towards pursuit in a setting that involves highly constraining roads and buildings. Thus, most of the environment and other agents are not visible to pursuing agents, making the utility of collective movements unclear. However, it still seems reasonable to expect that some amount of inter-agent communication and coordination will provide a benefit and raises the issue of whether recall of local road/building locations can facilitate team efforts.

In this context, we examine two hypotheses: first, that coordinated collective movements can still contribute to improving team performance as would be expected, and second, that giving individual agents even a very limited working memory provides benefits in the context of ubiquitous obstructions to movements and very limited agent visibility. For the first hypothesis, we compare the performance of coordinated versus independent movements by team agents. Since agents cannot directly see one another, coordination of movements is brought about by local broadcasting of information. For the second hypothesis, we compare situations where agents have limited local memory (working memory) versus global cumulative memory (episodic memory). Assessing these two hypotheses is important not only for theoretical reasons and intellectual curiosity, but also because of its practical importance in

contemporary work with semi-autonomous robotic teams. For this reason, our agents simulate physical robots in obtaining information about the local environment from a sequence of eye-level first-person images that must be interpreted by the agents as they move through the city.

Methods

Scenario, Environment, and Agent Specifications

The specific pursuit scenario considered here uses five agents and a single moving target that the agents are trying to capture. The agents move in real-valued space, but there is a discrete 32 by 32 grid of environmental features overlaid on this space (Figure 1 shows a sample city grid populated with the agents). Because agents exist in both real-valued space and grid space, they can potentially occupy the same cell as one another, but are restricted from passing through one another since each agent occupies space in the environment. The scenario ends when one of two criteria is met. If the time counter has reached 5000 time steps, the scenario ends in failure for the agents and success for the target. At any point before that, if the agents manage to surround the target such that there is either a building or agent within one cell north, south, east, and west of it, then the target is considered “captured” and the scenario ends with success for the agents and failure for the target.

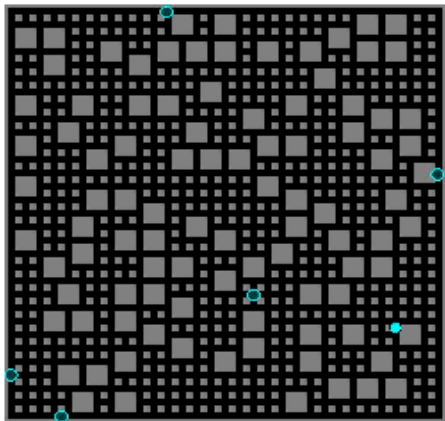


Figure 1: Sample urban environment map. The light-filled circle represents the target, the dark-filled circles represent the five agents seeking to capture it, gray regions represent small and large buildings (environmental obstacles that obstruct agent vision), and black regions represent streets.

As illustrated in Figure 1, the city consists of a grid of roads and buildings, along with a perimeter that is an unbroken wall of buildings leading to a contained setting. Given that each agent and the target use a first person perspective, they cannot see through buildings but can potentially see any distance along a road. Figure 2 displays a snapshot that is a typical example of one agent’s view. As illustrated here, an agent has a relatively narrow view angle (30 degrees). The agent receives the raw image shown on the left and processes the image using self-organizing maps, trained a priori, to segment the scene as shown on the right in

The Role of Working Memory in an Urban Pursuit Scenario

Figure 2 into buildings (marked “B”), roads (marked “R”), and the target object (marked “O”). Agents and target are different colors to allow for simple recognition of the target. Contiguous blocks of cells of the same type have been outlined in white in the right image. The image resolution produced by the “camera” is 128 by 128 pixels, and the scene-segmenting grid on the right has a resolution of 8 by 8 pixels per cell. When the agents move through the environment, they advance at a rate of 0.25 steps per unit time, while the target moves slightly faster with a step rate of 0.26.

Each set of trials for an agent team consists of 100 separate runs of the scenario (each with an allowed maximum of 5000 time steps). While the setup for individual runs differs from one another, the same set of trials is used for each of the different agent teams tested. Additionally, the agents and the target are not placed completely randomly in the environment. The target always begins in the center of the map, and one agent is always placed so that it can see the target from the beginning. This eliminates the need for the agents to find the target from the beginning and allows the scenarios to run more quickly. The other four agents are each placed randomly within the four different quadrants of the city, forcing them to be spread out at the beginning. None of these other agents begin the scenario seeing the target.

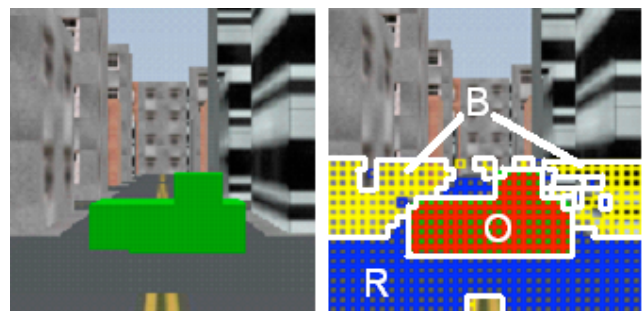


Figure 2: Sample snapshots of the pursuit environment from the perspective of an agent.

As the agents on a team receive a sequence of images of the environment as input, they follow a pursuit strategy that determines how well the team performs. Different agent teams use different strategies. These different strategies naturally show a difference in performance as certain features are included or excluded. The features that were varied between teams were communication between the agents, type of memory strategy used, and the use of movement coordination. While the strategies used by different teams varied, the target’s behavior remained the same in all cases.

General Movement Strategies

Movement in the environment occurs for every agent at every time step, with a new position chosen given the agent’s rate of movement and current velocity vector. Agents can move reflexively to stimuli in the environment or can place waypoints to direct their future movement towards a specific observed location, usually a point where the agent will want to make a turn. The agent is frequently required to estimate the locations of objects in the environment from its first-person view image. This is done by examining pixels along the base of a seen object. Using its height and angle of view in the

vertical dimension, an agent can estimate the distance of any row of pixels in the snapshot. Using this information and the distance from the central vertical line of the snapshot, the agent is able to estimate the location in the environment.

In the simulations studied here, we are interested in assessing the relative value of agents having only a local memory (recall only information that is local in space and time) relative to agents having a cumulative memory (recall information seen since the start of a simulation. Very roughly speaking, local memory corresponds to human working memory with its very limited capacity (Cowan et al, 2005), while cumulative memory corresponds to human episodic memory (Tulving, 2002).

Accordingly, agents in a team have one of two types of basic memory available to them: local memory only (LM), or cumulative memory (CM). LM agents have a memory of what is seen in the surrounding environment, and convert some of this information into retained knowledge of the surrounding space for navigation. Thus, in addition to using the knowledge of the environment from the view window at every time step, an LM agent also keeps a local memory of its immediate surroundings. This is an $n \times n$ two-dimensional map designed as a cellular space, built by and continually updated by the agent as it moves, where $n = 3$ in the current pursuit scenario; see Figure 3. This small map is centered on the agent, but aligned with the global grid. Each cell is given a different value depending on what the agent has estimated exists in that cell from its recently observed visual information.

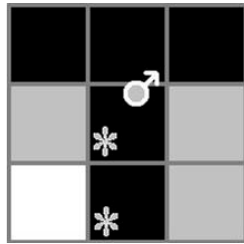


Figure 3: An example local memory map produced by a LM agent in the simulated environment, forming a three by three grid. The agent is represented by the circle with a directional arrow in the central cell. Black cells are cells estimated by the agent to be streets. Gray cells are places estimated by the agent to be buildings. White cells are unknown, meaning the agent has not acquired information about them. Black cells marked with an asterisk are areas the agent remembers passing through. As the agent moves, the center of its local memory map moves with it.

As in creating waypoints when an agent has no memory, some conversion of first-person perspective is required by LM agents to create a 2D local area map. Pixels and their locations in the environment are computed in the same way as with agents having no memory, but what is done with the information is different. For each pixel along the terrain level (approximately the lower half of a snapshot view), the location of its cell is determined relative to the agent. If this estimated location falls in a cell of the agent's local memory, then it contributes to generating a memory of the environment surrounding the agent.

The contents of LM cells (Figure 3) are determined by the agent to be in one of four possible categories: unknown (if the agent has not seen the cell's contents), visited (if the agent has been in the cell before it leaves its local memory), passable (if the cell is perceived to contain predominantly streets), and impassable (if the cell is perceived to contain predominantly buildings). These determinations shift as the agent moves through the environment. In this way, when an agent moves into a cell it has previously recognized as "passable" in its LM (i.e., when this passable region becomes a "visited" region), then all of the information currently in memory shifts to the appropriate cells, while some old information is lost for those cells that are no longer close enough to the agent, while new "unknown" areas may appear in those cells that have now become close enough.

The second CM (cumulative memory) strategy makes use of the same techniques as those of the LM agents, but it adds the ability for the agent to have a global map that it constructs from its cumulative LM memories over the course of a simulation. This global map is used to generate a sequence of planned movements from its present location all the way to an intended destination. This method makes use of much more information than the previous LM method. An example global map built by an agent is depicted in Figure 4.

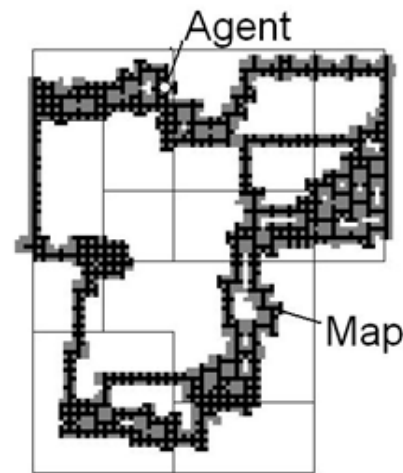


Figure 4: Representation of an example cumulative memory map produced by a single agent after touring parts of a simulated environment. Black cells are places estimated by the agent to be streets. Gray cells are places estimated by the agent to be buildings. White cells are unknown, meaning the agent has not acquired information about them. The map also includes lines denoting district boundaries of the city, which are known to the agent and can influence agent behaviors.

Like with the local memory used by LM agents, the global map constructed by a CM agent is a cellular space aligned with the world's grid. It is calibrated to be the same size as the world. It is not centered on the agent, but the agent moves through it and knows its own location at any point in time. This means the agent, aware of its own global position, is able to compute the appropriate global map changes in the cells near it, often changing nearby cell states from "unknown" to other states, as it moves through the environment. A CM agent that selects a target destination is able to plot out a sequence of waypoints across the map that will allow it to

reach the target location. The planned movement path may include cells with both known and unknown content. An A* search algorithm is used to generate planned movements, where the search space is the agent's grid map, the start state is the agent's current cell, and the goal state is either a specific cell or any cell in a specified district. In generating a planned route, areas known to be impassable are excluded from the route, but known passable areas do not get priority over unknown areas, because there may be more direct routes that pass through unknown territory. This planned movement route composed of waypoints is repeatedly updated.

Agent and Target Behaviors

The mobile target and the pursuing agents have general behaviors that are programmed into each of them at the beginning of a scenario.

The target has two behaviors: patrol the city (i.e., move arbitrarily throughout the city) and evade pursuing agents. Patrolling behavior occurs when the target cannot see any pursuing agents, while evasion occurs whenever a pursuing agent is observed. Unlike the pursuing agents, the target behavior rules are the same across all test runs: use a local memory strategy while patrolling, and a cumulative memory strategy while evading. The target's memory of the environment, when using a cumulative strategy, is unlimited, while its memory of obstacles is temporary since agents rarely stay in the place the target saw them for long. The target is always moving and attempting to get out of sight of its pursuers. The target is oblivious to any communication between agents and cannot communicate with them.

The agents have two alternative behaviors: patrol the city or pursue the target. Patrolling the city occurs when an agent is unaware of the target's location. It is a high-level behavior where the agent moves about the city to cover as much unseen territory as possible. A CM agent will therefore try to move into areas it has never explored previously, while LM agents move randomly because they forget where they have previously visited. Finding the target causes an agent to announce it has seen the target when the target is first visible, and to begin pursuit.

In simulations where agents are given the ability to communicate, when an agent sees the target it will broadcast the target's location to all other agents within a broadcast range (15 grid cells). The agent will also adjust its normal patrol behavior, following the target in a manner depending on which memory strategy is in use (described below). The agent also remembers seeing the target, so if the target should turn onto a side street, the agent will continue to the last location where the target was seen and then turn to match the last remembered angle of the target. If this succeeds in returning the target to view, the agent will continue this process. If it fails, then the agent returns to a regular patrol of the environment, looking for the target it lost, unless it receives a broadcast from another nearby agent about the target's current location.

When an agent receives a broadcast of the target's location, if the agent receiving the broadcast does not also see the target, then it no longer behaves as described above. Instead, the agent treats this as a command to go to the broadcast location to assist in trapping the target. As the broadcast

The Role of Working Memory in an Urban Pursuit Scenario

updates, so too does the goal location of the agent hearing the broadcast. It will then move depending on the agent memory strategy in use for this scenario. The planned movement of other agents and remembered portions of the environment's layout may also affect how the agent moves, as follows.

When an agent uses a local memory strategy, the method of pursuit is simple. If the LM agent can see the target, the agent computes the direction of the target and places a waypoint in a position in the environment that will guide it toward the target. If the agent cannot see the target, but saw it recently and remembers the target's last known location and direction, the agent will continue in the process of generating and approaching waypoints toward that location until it is less than half a cell away. When the agent reaches this threshold, it matches the target's last known direction angle in order to achieve the best chance to see if the target is still visible and to continue on the same heading that the target last had.

If the LM agent does not see or remember the target, but receives a broadcast of its current location, the agent will head to that location. The agent places a waypoint in the closest adjacent cell that takes it in the direction of the broadcast target location. As the target's position is updated in the received broadcast—assuming another agent sees it—then the agent receiving the broadcast will continue to place and approach waypoints that move it closer to the changing target location until it either sees the target or moves out of range of the broadcast. If the broadcasts cease, but the agent remembers a broadcast target location, it will continue to approach that location until a new broadcast is issued or the agent sees the target.

When the agent is using a CM strategy, the method of pursuit differs. If the agent has the target in view, it computes a path of waypoints to the target's position and follows them. If not, but the agent recently had the target in view and remembers its last known location and direction, the agent generates a path of waypoints to that location. When the agent reaches the last seen position of the target, it matches its last known direction angle. If the CM agent does not see or remember the target, but receives a broadcast of the target's current location, the agent generates a path of waypoints to that location that it will follow. Because the broadcast target location the agent receives will potentially update as an announcing agent tracks the target through the environment, the waypoint paths generated by the CM agents relying on the broadcast information are also updated. Again, if the broadcasts cease, but the agent remembers a broadcast target location, it will continue on the path to that location until a new broadcast is issued or the target enters its view.

Inter-Agent Communication and Coordination

Basic agent communication consists of broadcasting the estimated location of a target, when the target is visible, to all other agents within its broadcast radius. While not a completely accurate position of the target, especially if the agent is far away from the target, this broadcast position can still point other agents within the broadcast range toward the correct general area, giving them a greater chance of finding the target and making their own broadcast.

In more open environments, coordination between agents is usually achieved through accelerations based on direct

observation of movements of other nearby agents, leading to collective movements (i.e., to agents moving in a “flock”). In contrast, due to the cramped nature of this environment, where agents are less apt to be able to see other agents, such coordination is less likely to prove beneficial. This is particularly the case with the very limited view angle of each agent. This view angle, when coupled with the many building obstacles, means that the agent can effectively see only straight ahead along the road they are using.

Nevertheless, avoidance acceleration remains useful in this scenario in situations where the agents might collide and have difficulty navigating around one another. It also helps to separate them, allowing them to spread out more and cover more of the environment. The radius of this avoidance influence is typically kept quite low (one cell length). Because this could interfere with the pursuit of a target, the avoidance influence is only factored in when the agent does not see the target and is not approaching the last place it remembered seeing the target. If the agent is close enough to another agent to experience this influence, then it alters its course to a waypoint that is in a valid adjacent cell that is closest to its avoidance vector. The avoidance vector is computed such that the agent only avoids the closest neighboring agent in its avoidance radius.

In addition to this influence, there is another method of agent coordination that can be implemented in the case of a cumulative memory strategy. The chances of a successful outcome are increased if the agents are approaching the target from different directions; this increases opportunities to surround and thus capture the target. If the paths can be coordinated so they cross as little as possible, then the agents’ performance should improve. This can be achieved by having an agent broadcast its planned path when the agent has planned a path to the target’s location, whether it was seen, remembered, or received through a broadcast. When planning a path, the cost of waypoints increases when a location is on another agent’s path as well, making it more likely the agent will attempt to find another direction from which to approach the target. Additionally, the cost of any cell at the end of another agent’s path that is adjacent to the broadcast target’s current location has an even higher cost, increasing the chances that agents will look for valid paths that approach the target from another direction.

Results

In the baseline scenario, there was no communication between agents and therefore no coordination of agent positions or movements. Pursuing agents may visually perceive one another in the environment, but this does not influence their behavior. The agents use a local memory strategy for both their patrol and their pursuit behaviors. Given the limit of 5000 time steps to complete the task of capturing the target, the agents were successful in 70% of the trials (see Figure 5). The mean completion time for this set of simulations was 2904 time steps. This indicates that while agent teams can often solve the problem while the agents move independently, there is still room for improvement in that close to a third of the time they are unsuccessful.

The Role of Working Memory in an Urban Pursuit Scenario

We varied the agent behaviors away from baseline in a variety of ways to observe the effects. In a second scenario, agents were given the ability to communicate (Figure 5). Once communication is enabled, agents that see the target broadcast its location, and other agents that hear it have some knowledge of where to intercept the target. This is the only difference between this scenario and the baseline scenario, with the communicating agents experiencing a 97% success rate and a mean completion time of 804 time steps, which is a significant improvement (a paired t-test gives $p < 0.05$) over the original mean of 2904. A slight further improvement in the mean completion time is also gained when a small agent avoidance influence of radius 1 is also included in this scenario (see Figure 5); accuracy rises to 98%, and the mean completion time drops to 744. These three results support our first hypothesis that agent cooperation/communication continues to be effective even in the context of densely occluded movement spaces like those used here.

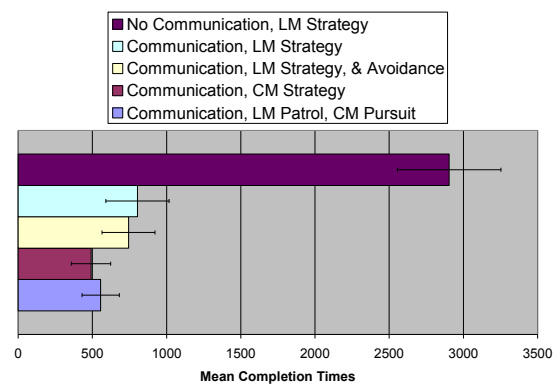


Figure 5: Mean completion times for agent teams using several different strategies as described in the text. The worst scenario was the baseline LM strategy where no communication occurs between agents (70% success rate). There is significant improvement whenever agents are given the ability to communicate. There is further improvement when the agents adopt a cumulative memory strategy for either all of their behaviors or just the pursuit behavior. The error bars here represent 95% confidence intervals.

The three scenarios described so far have used solely local memory strategies for both patrol and pursuit behavior. Either or both of these strategies can be changed to using a cumulative memory strategy. While a cumulative memory strategy would not be expected to improve patrol behavior significantly, it should improve the performance of pursuit, because agents in this latter mode are actively trying to get somewhere as quickly as possible, as opposed to simply exploring. When the cumulative memory strategy is applied to both patrol and pursuit behaviors in the context of communicating agents, the success rate increases to 99% and its mean completion time decreases to 491 time steps (Figure 5). When the cumulative memory strategy is applied to only the pursuit behavior, all trials successfully completed before the time limit, and the mean completion time was 556 time steps. Both of these mean completion times are a significant improvement over the local memory strategy, but do not have a significant difference from one another. As expected, a

cumulative memory strategy was able to improve upon a strategy where agents were able to acquire less information. However, it is apparent that this is also useful in a scenario where multiple agents are performing a task together. Figure 5 also shows these additional results. The results are consistent with our second hypothesis that even a use of local memory can substantially improve agent performance. As seen in Figure 5, agents with local memory improve almost as much as those with cumulative memory relative to baseline.

To further examine whether coordination is a useful feature for pursuing agents in densely occluded environments, the scenarios that used cumulative memory were also tested to see what improvement could be gained from introducing an avoidance influence or from allowing agents to broadcast their planned movement paths so that they could coordinate in an attempt to surround the target. Figure 6 shows the results for these scenarios.

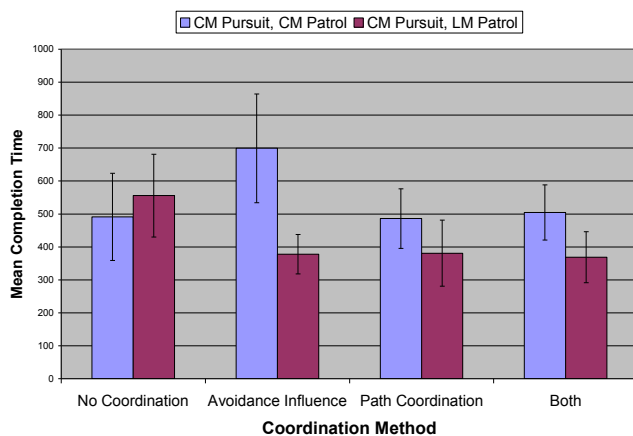


Figure 6: Mean completion times for the different coordination methods in scenarios with different memory strategies involving cumulative memory. In 100% of these simulations the pursuing agents completed the task within the time limit except for the cumulative memory pursuit and patrol strategy with only the avoidance influence, where 99% of the tasks were completed. The error bars are 95% confidence intervals.

The results here differ substantially depending on the memory strategy used. In the scenarios where cumulative memory is used for both of the two major agent behaviors (pursuit and patrol), adding these coordination methods, either had almost no effect or made the performance significantly worse (as in the case with an avoidance influence). However when a local memory strategy is used for the patrolling behavior, both coordination methods and their combined use yield a significant improvement. Thus the benefit of these coordination methods appears to be mitigated when using a cumulative memory patrol. For path coordination, this is likely due to interference between patrol paths and pursuit paths, which unexpectedly caused agents to influence each other even in situations where there is no obvious benefit from this interaction. The avoidance influence also evidently interferes with the patrol behavior, probably because the agents create distant goals for themselves when patrolling using cumulative memory. When paths cross, it is more

difficult for two agents with different goals to reconcile them with just this influence. This situation is less likely to occur when agents are in pursuit of the target, because if they are in close proximity to one another, they are likely heading in the same direction and do not have conflicting paths.

The above results demonstrate that there is usually a large benefit from allowing communication between pursuing agents, a substantial benefit from allowing coordinated agent behaviors, and a substantial benefit from using a memory strategy with more information. However, path coordination relies on the use of cumulative memories, which substantially determine the details of the planned paths. Is the benefit seen with path coordination due mainly to the coordination, or does the inherent presence of memories, influencing the paths, also significantly contribute to the improvement? This issue was addressed by comparing a scenario that features the minimum amount of memory necessary for the agents (i.e., knowing the contents of four cells, the cell the agent currently occupies and the three additional cells in front of them), against scenarios with increasing memory capacities, all of which use path coordination.

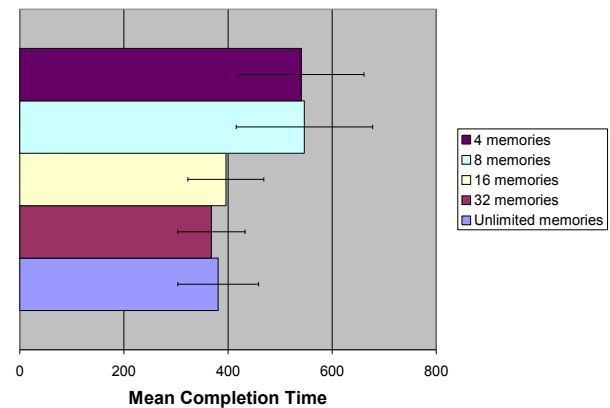


Figure 7: Mean completion times for the scenarios featuring limited memory (4 memories = contents of 4 cells recalled, etc.). Not included in this chart are the results where the agent has a limit of no memories (i.e. remembered cells). These latter agents perform much worse, with below 100% accuracy and a mean completion time in the thousands. The error bars are 95% confidence intervals.

A scenario was tested in which agents used local memory strategies for patrolling and cumulative memory strategies with path coordination for pursuit, but with a requirement that an agent remove a random old memory cell when adding a new memory based on visual data when the memory capacity is exceeded. This effectively kept a limit on the size of an agent's memory, although memory contents could change and be updated as the agent moved through the environment. During this test, as memory capacity was increased by a factor of two, the mean time to success dropped. By the time the threshold for removing old memories is 32, there was a significant improvement (a paired t-test gives $p < 0.05$) over the scenario with minimum memory capacity, and no significant difference between its performance and that of the scenario with unlimited cumulative memory. These results,

depicted in Figure 7, again demonstrate that a limited, incomplete memory of the environment can be very effective.

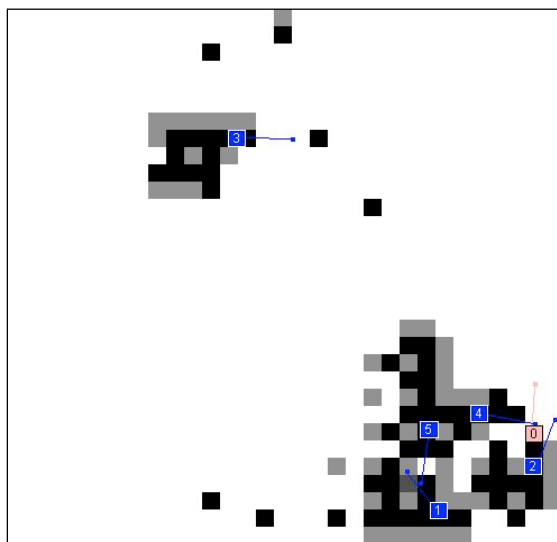


Figure 8: Composite depiction of the local maps constructed by each of the agents in the environment. An individual agent's local memory capacity limit is 24 in this case; what is shown here is the memory of the entire agent collective, which is effectively the "sum" of the individual agent memories. The target's memory, which is always unlimited, is not displayed. Black areas are streets remembered by at least one agent, gray areas are buildings remembered by at least one agent, and white areas are unknown to anyone. Some areas that are in between the normal shades of black and gray are places where the agents have differing memories of what is there. Agents are indicated by numbered boxes outlined in white. The box numbered 0 is the target. The agent numbered 2 is pursuing the target, and agents 1, 4, and 5 are in agent 2's broadcast range.

Finally, we consider the question of why agents with just a local memory perform almost as well as those with cumulative memory. Figure 8 displays a representative composite map at a single time step of the remembered cells of agents with limited memories. Even though agents do not tend to remember distant locations given the frequent updates of memory, the coordination of paths in the collective memory strategy has an impact because the agents are often trying to find ways to effectively spread out and surround an area they have recently explored and where they are told the target is estimated to be located. Knowing more than a few features is very useful for this, and shows that a memory strategy somewhere between highly local and completely cumulative is almost as effective as the fully cumulative CM strategy.

Discussion

In terms of coordination, we found that agents without any type of communication or coordination had far worse

The Role of Working Memory in an Urban Pursuit Scenario

performance than agents where communication about the target location and various types of coordination (collision avoidance, planned movement path overlap avoidance) were introduced. While not consistent in improving performance, both types of coordination did make significant improvements in situations where two different memory strategies were used for pursuit and the patrol. While this is not surprising with the path coordination, which was designed so the agents would tend to surround the target, it is surprising that the avoidance influence had such a significant impact. It should be noted that when the radius of the avoidance influence is increased, the effect is detrimental, so avoidance proved effective mostly as a means of ensuring agents did not get stuck when they collided with one another.

With respect to agent memory, there are three main conclusions to emerge from the computational experiments done with this work. First, the results support the hypothesis that adding an individual working memory of the obstacles encountered by an agent team can significantly improve both its success and efficiency in accomplishing the pursuit scenario. This occurred even in this simulated environment where agent input was limited to a sequence of images from a first-person perspective, and even when the agent team was already benefiting from communication and coordination. Second, the results also support the hypothesis that even when the size of an individual's working memory is severely limited, the agent team as a whole still experiences a significant improvement in its efficacy and efficiency. Third, the results indicate that a different memory strategy for different tasks works best for the agent team. Local memory is found to be better for the patrol behavior, while a cumulative memory strategy is found to be better for the pursuit behavior. As expected, granting the agents a local communication, so that they could broadcast the estimated location of the target, improved performance greatly, allowing agents to converge on the target more easily.

More interesting than the general advantages of the cumulative over the local memory strategy is the fact that a combination of strategies tended to work quite well. In cases with coordination, it worked much better for the pursuing agents, where they would patrol using a local memory strategy, but pursue the target with a cumulative memory strategy. This gave them flexibility in searching, while allowing them to make decisions, potentially informed by memories, when trying to quickly reach the target's location upon hearing a broadcast.

Of all the observations in this study, perhaps the most important result is the finding that giving individual agents even a limited memory of the environment could give the agent team as a whole a significant improvement in performance, even when improvements were already present due to communication, path generation, and coordination. What was surprising about this is that the individual agent's memory capacity can be so small, because in viewing the agent team as a system, the resulting collective memory consists of the memories of all its agents combined with one another, albeit distributed between individuals (as illustrated in Figure 8). This is why agent teams with local memories seemed to do so well, and it suggests that much of the information stored by cumulative memory agent teams was of

very limited usefulness. These results are consistent with those obtained in an earlier study with a much simpler environment and much simpler agents (Winder and Reggia, 2004).

Acknowledgement: Supported in part by funding from the United States Government.

References

- Alcazar, J. (2004). A simple approach to the multi-predator multi-prey domain. In *International Conference on Complex Systems*, Boston.
- Atherton, J., Hardin, B., and Goodrich, M. (2006). Coordinating a multi-agent team using a multiple perspective interface paradigm. In *Proceedings of the AAAI*, pages 47-51.
- Bayazit, O., Lien, J-M., and Amato, N. (2002). Roadmap-based flocking for complex environments. In *10th Pacific Conference on Computer Graphics and Applications*, pages 104-113.
- Benda, M., Jagannathan, V. and Dodhiawalla, R. (1986). On optimal cooperation of knowledge sources, Technical Report, Boeing Advanced Technology Center, Boeing Computer Services, Seattle.
- Couzin, I., Krause, J., Franks, N., and Levin, S. (2005). Effective leadership and decision-making in animal groups on the move. *Nature*, 433:513-516.
- Cowan, N., Elliot, E., Saults, J., et al. (2005). On the capacity of attention. *Cognitive Psychology*, 51:42-100.
- Hladek, D., Vascak, J., and Sincak, P. (2009). Multi-robot control system for pursuit-evasion problem. *Journal of Electrical Engineering*. 60(3):143-148.
- Huang, F., Wang, L., Wang, Q., Wu, M., and Jia, Y. (2009). Coordinated control of multiple mobile robots in pursuit-evasion games. *American Control Conference*. pp. 2861-2866.
- Kennedy, J., Eberhart, R., and Shi, Y. (2001). *Swarm Intelligence*. Academic, San Diego, CA.
- Kurihara, K., Nishiuchi, N., Hasegawa, J., and Masuda, K. (2005). Mobile robots path planning method with the existence of moving obstacles. In *10th IEEE Conference on Emerging Technologies and Factory Automaton, 2005*, pages 195-202.
- Lapizco-Encinas, G., Kingsford, C., and Reggia, J. (2009). A cooperative Particle Swarm Optimization Method for Side-Chain Packing. *Proc. IEEE Swarm Intelligence Symposium*.
- Lenzitti, B., Tegolo, D., Valenti, C. (2005). Prey-predator strategies in a multiagent system. *Proceedings of Seventh International Workshop on Computer Architecture for Machine Perception*, pages 184-189.
- McCook, C., and Esposito, J. (2007). Flocking for heterogeneous robot swarms: a military convoy scenario. In *39th Southeastern Symposium on System Theory*, Macon, GA.
- Nuxoll, A. and Laird, J. (2011). Enhancing intelligent agents with episodic memory. *Cognitive Systems Research*, In Press. doi:10.1016/j.cogsys.2011.10.002
- Reynolds, C. (1987). Flocks, herds and schools. *Computer Graphics*, 21:25-34.
- Rodriguez, A. and Reggia, J. (2005). Collective-movement teams for cooperative problem solving. *Integrated Computer Aided Engineering*, 12:217-235.
- Rodriguez, A. and Reggia, J. (2009). A distributed learning algorithm for particle systems. *Integrated Computer-Aided Engineering*, 16:1-20.
- Tulving, E. (2002). Episodic memory: from mind to brain. *Annual Review of Psychology*, 53:1-25.
- Vail, D. and Veloso, M. (2003). Dynamic multi-robot coordination. *Multi-Robot Systems*, pages 87-100.
- Winder, R. and Reggia, J. (2004). Using distributed partial memories to improve self-organizing collective movements. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(4):1697-1707.
- Zhao, D., Jin, W. (2005). The study of cooperative behavior in predator-prey problem of multi-agent systems. In *Autonomous Decentralized Systems. ISADS 2005. Proceedings*, pages 90-96.

Behavior and Intelligence

Papers

Second Order Learning and the Evolution of Mental Representation

Solvi Arnold¹, Reiji Suzuki¹ and Takaya Arita¹

¹Graduate School of Information Science, Nagoya University, Aichi 464-8601, Japan
solvi@alife.cs.is.nagoya-u.ac.jp

Abstract

Mental representation is a fundamental aspect of advanced cognition. An understanding of the evolution of mental representation is essential to an understanding of the evolution of mind. However, being a decidedly mental phenomenon, its evolution is difficult to study. We hypothesize how interactions between adaptation levels may cause emergence of isomorphism between a cognitive system and its environment, and that mental representation may be understood as an instance of this effect. Specifically, we propose that selection for second order learning translates into selection for isomorphism-based implementation of first order learning ability, and that mental representation is (an aspect of) the environment-cognition isomorphism produced by such learning ability. We then give a reformulation of cognitive map ability, a paradigm case of mental representation, in terms of our hypothesis and explore it computationally by evolving a neural network species with the neural basics for second order plasticity (the basis for second order learning) in an environment composed of randomly generated maze tasks, including tasks generally believed to require mental representation (in the form of cognitive maps). The model is shown capable of evolving nets that solve these tasks, providing preliminary support for our hypothesis.

Introduction

Mental representation (MR for short) is, abstractly put, the ability to simulate or reconstruct in the mind aspects of the environment that lie outside the scope of one's current perception. The type of MR we focus on in this paper is the ability to navigate complex environments using "cognitive maps": mental representations of the spatial layout of an environment (see Tolman 1948). Cognitive maps aid navigation, as typically only a limited part of the area to be navigated is directly perceptually accessible. Other types of MR are "mental time-travel" and "theory of mind" (see Takano & Arita 2006, Minoia et al., 2011 for computational approaches to the latter). There too, inaccessible aspects of the environment (respectively: future and past, other minds) are mentally simulated or reconstructed.

The evolution of MR is not well-understood. MR is a highly structured and organized form of cognition, and already in the early decades of connectionism, it has become clear that (contrary to common intuition) adaptive processes such as evolution or learning do not, in general, produce such structured or organized AI (see e.g. Fodor & Pylyshyn, 1988). If our simulated adaptation processes do well at producing non-representational cognition, but fail to produce representa-

tional cognition, then this raises the question how MR can have evolved in biological cognitions. The question seems particularly important since there appears to be a tight conceptual link between representation and intelligence. Non-representational cognition can be attained via what we might call blind adaptation, be it mutation and selection fashioning fit but fixed innate behaviour, or trial-and-error learning chasing a reward-signal. Representational cognition, or at least the behaviour we recognize it by, is characterized by more advanced forms of adaptation. We recognize intelligence by the *absence of trial and error*: a solution is mentally represented, then executed. *Insight* crucially depends on representation.

We propose the following explanation of evolution of mental representation ability: As learning ability evolves, the need for trial-and-error is reduced. This reduction is attained by adapting to the environment the process that adapts behaviour to the environment, that is, by second order learning. Selection pressure on second order learning translates into selection pressure on isomorphism-based implementation of first order learning. Mental representation is part of this isomorphism.

The idea of a central role for isomorphism in the evolution of cognition is not new: Herbert Spencer viewed the evolution of mind as ever expanding correspondence between the internal and external (Spencer 1855, see also Godfrey-Smith 1996). Our contribution is a hypothesis on how evolution and the orders of learning interact to produce such correspondence.

We provide a proof of concept for our hypothesis in the form of a computational model in which a neural network species with the basic constituents for second order plasticity (the neural basis for second order learning) is evolved in an environment containing maze tasks generally believed to demand cognitive map ability. The model is shown capable of evolving nets that solve these tasks, providing preliminary support for our hypothesis.

Isomorphism & Learning

Our theory explains MR as an instance of a more general organization effect. In this section we explain this effect, and in the following sections we discuss how it applies to MR. We first define our main terms:

Behaviour: a mapping from stimuli (S) to responses (R).

$$L_0: S \rightarrow R$$

We denote behaviour as L_0 because in our theoretical framework it occupies the position of zero-order learning.

1st order learning: given the current behaviour and a stimulus, updates the behaviour.

$$L_1: (S, L_0) \rightarrow L_0, \text{ i.e.:}$$

$$L_1: (S, (S \rightarrow R)) \rightarrow (S \rightarrow R)$$

2nd order learning: given the current 1st order learning map and a stimulus, updates the 1st order learning mapping.

$$L_2: (S, L_1) \rightarrow L_1, \text{ i.e.:}$$

$$L_2: (S, (S, (S \rightarrow R))) \rightarrow (S \rightarrow R) \rightarrow ((S, (S \rightarrow R)) \rightarrow (S \rightarrow R))$$

And so on for higher orders, though we do not concern ourselves with anything above L_2 here.

Environment: a mapping from responses to stimuli:

$$E: R \rightarrow S$$

Note that an environment is much like an inverse behaviour (mapping responses to stimuli instead of stimuli to responses).

Each mapping may additionally update internal states, but for simplicity we leave this out in notation as we do not need it for this explanation.

We have defined mappings, but organisms are physical objects, not mathematical objects. In order for these mappings to exist in the physical world they must have implementations. For each of the mappings defined above, we let its lowercase partner denote its implementation: l_0, l_1, l_2, e . Implementation of the environment mapping here should be understood as the actual physical reality of the environment.

An organism's fitness depends on the mappings it implements, but generally not on *how* it implements those mappings. Any given mapping can be implemented in infinitely many ways (indirect fitness effects such as energy cost may weed out overly unwieldy implementations, but still leave many viable options). This poses a problem for understanding the evolution of mind: mental phenomena are part of the implementation of our mappings, but evolution does not generally care how we arrive at our responses as long as they fit the stimuli that triggered them. Why *this* implementation, and not another?

How real of a problem this is can be clearly seen in the history of the philosophy of artificial intelligence: the problem of connectionist systematicity may be interpreted as the problem that artificial adaptation processes typically fail to pick a systematic implementation from the set of viable implementations that solve the problem set they run their adaptation process on. Highly diffuse and unorganized implementations are viable for surprisingly complex tasks.

Yet we feel quite sure that cognition is the product of evolution, and given how systematic and seemingly organized it is, it seems unsatisfactory to appeal to coincidence as cause for this particular implementation. The question of what factors guide implementation choice in evolution is essential to an understanding of the evolution of mind, but remains largely unanswered.

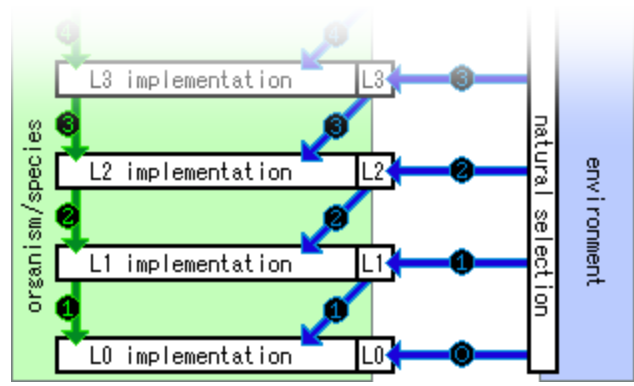


Fig. 1. Evolution and the orders of learning. Green arrows indicate adaptation on within-lifetime timescales (i.e. learning), blue arrows indicate adaptation on evolutionary timescales. As indicated by the diagonal blue arrows, selection operates only indirectly on the implementation structure of any given adaptation level, via the effect that implementation structure has on the feasibility of the adaptation level above it. In general, direct selection for adaptation level i converts into indirect selection for isomorphism-based implementation of adaptation level $i-1$.

Figure 1 expresses the relations between evolution and the orders of learning (including behaviour, as L_0). Each order of learning L_i ($i > 0$) adapts L_{i-1} (the green downward arrows) on a within-lifetime timescale. We pick any two adjacent orders of learning (L_0 and L_1 in (Arnold, 2011), L_1 and L_2 in most of this paper). If the environment has the sort of dynamics to which L_i is applicable, then there is selection pressure on evolution of L_i . Different implementations of L_{i-1} call for different implementations of L_i . For example in the highly unnatural case that l_{i-1} would take the form of a table defining an output for each possible input independently, then l_i would operate by rewriting entries of this table. So whether and how feasible evolution of L_i is strongly depends on l_{i-1} . If there is selection pressure on L_i , then mutations in l_{i-1} that are beneficial to L_i are beneficial mutations (even if they have no effect whatsoever on L_{i-1}). As an extreme scenario, we could imagine L_{i-1} remaining stable while l_{i-1} evolves to facilitate L_i . This possibility shows that there is a fundamental difference between selection for a specific mapping and selection for a specific implementation of that mapping.

So while evolution working on L_{i-1} alone does not care much about the structure of l_{i-1} , "co-evolution" (if we may abuse the term a little) of L_i and l_{i-1} *does* care about the structure of l_{i-1} . Along the horizontal blue arrows in figure 1, evolution treats its objects as black boxes (selecting on input-output relations alone), but indirectly through the neighbouring learning order above it (diagonal blue arrows), it peeks inside and selects for implementation structure.

L_i constrains l_{i-1} , but we haven't said anything yet about what sort of l_{i-1} is favoured by L_i . We will claim that L_i benefits most from l_{i-1} s that are in some sense isomorphic with the environment. For the simplest case, L_0 and L_1 , the basic idea is as follows: If the environment and (consequently) the optimal behaviour are static, then difference in the structure of their implementation poses no problem. But if the environment and (consequently) the optimal behaviour may change (by means of L_1), then the more the structure of l_0 and e differ, the harder it is for L_1 to update L_0 in sync with E . The implementations

(e) of environments that cognition evolves in are composed of distinct aspects (food sources, temperatures, other agents, spatial layouts, etc. etc.) that act and interact to give rise to E . Let's call a change in one such aspect a *simple* change. Simple changes in e often lead to *complex* changes in E : multiple input-output pairs change. Consequently a complex update of L_0 is required. If l_0 contains an aspect corresponding to the changed aspect of e , in a functionally similar position, then the required complex change in L_0 can be realized by a simple change in l_0 . This makes L_1 quite feasible. If no such corresponding aspect exists, a complex implementation update is required. In this case no straight-forward relation exists between the environmental change and the appropriate behaviour change, making L_1 's work difficult or infeasible.¹

So the organization that evolves in l_0 to facilitate L_1 should in one form or another capture the variable aspects of the environment along with their functional roles therein. This correspondence is what we mean by isomorphism. Note that we do not claim that L_1 is strictly impossible without isomorphism between e and l_{i-1} , nor that such isomorphism cannot occur in absence of L_i . What we claim is that selection pressure on L_i translates into selection pressure on isomorphism at l_{i-1} , and that this selection pressure conversion is an organizing factor in the evolution of cognition.

Hopefully the argument for the case of L_1 and l_0 is clear now, but the focus of this paper is the case of L_2 and l_1 . The two cases differ most importantly across the type/token distinction between species/specimen, that is, the timescale on which isomorphism is acquired. Without L_1 , L_0 is static per specimen (for convenience we ignore other factors that modify behaviour). It is a given individual's innate behaviour. With L_1 redirecting selection onto l_0 , we should see *evolution* of isomorphism in l_0 , i.e. isomorphism in the innate organization of cognition (*innate isomorphism*). This effect was demonstrated in (Arnold, 2011). The topic of this paper, MR, may be described as isomorphism, but clearly it is not innate isomorphism. Mental representations are acquired on the within-lifetime scale, the timescale of learning. Mental representations are a form of *acquired isomorphism*, the result not of evolution processes but of learning processes.

What is the function of acquired isomorphism? Looking at Figure 1, we may hypothesize that, just as isomorphism evolved at l_0 benefits L_1 , so does isomorphism acquired at l_1 benefit L_2 . Evolution of isomorphism-acquisition at l_1 (which should include MR), then, would be a consequence of selection for L_2 .

¹ One might object that reinforcement learning algorithms manage to learn just fine without dependence on such "corresponding aspects". However, such algorithms depend on a reinforcement signal. We cannot in general assume such a signal to be available. Stimuli delivered by the environment may convey information about the fitness effects of a given response, but in natural settings the signal is more often than not incomplete, extremely noisy, or absent altogether. Even when a clear signal is available, a reinforcement learning algorithm must still depend on extensive trial-and-error learning to adapt to a complex change in E , even if the change in e is simple. Learning in biological species is routinely seen to do better than that, on basis of less information (e.g. first language acquisition).

This somewhat cryptic proposition will become clearer in the next sections, where we apply our framework to a well-known experiment from cognitive psychology in which the role of acquired isomorphism is intuitively clear, and show how that role can be understood in terms of second order learning.

Tolman's Detour Mazes

In experimental psychology, MR ability in biological species is often studied using Tolman's detour maze (Tolman & Honzik, 1930). These mazes have multiple paths (typically three) from their start to their goal, varying in length (see Figure 2). The shorter two paths join some distance before the goal position. The experiment runs as follows: a rat is fed to satiation, then placed at the start of the maze. A food reward is placed at the goal position. The rat explores the maze, and eventually finds the food reward, but, being satiated, does not eat it. After the rat has thoroughly explored the maze, it is taken out. We call this the exploration phase. Later, once the rat is hungry, it is placed again at the start position in the maze. The rat will now typically try to run the shortest path to the goal position and eat the reward. We call this the exploitation phase. In this phase, MR ability can be revealed by blocking the shortest path and observing the rat's reaction. If the shortest path is blocked such that the medium path is still open (in Figure 2: blocked at a cell with only a green dot) then the rat would ideally choose the medium path. If the shortest path is blocked such that the medium path is blocked too (in Figure 2: blocked at a cell with both a green and an orange dot), then the long path is the correct choice. If the rat, upon encountering the blockage, backtracks to the start position and then picks the new optimal path, then this taken as evidence of MR ability: If the rat had merely learned to solve the maze using action-sequences or state-action pairs, then finding one path blocked would tell it nothing about the viability of the other paths. So if it can pick the correct path right away, then it must also have grasped the spatial relations between the paths. That is, it must have a spatial representation of the maze. Note that we recognize MR here by the *absence of trial-and-error*: we would *not* ascribe MR ability to the rat if it would need to try the other two paths to figure out which choice is now optimal.

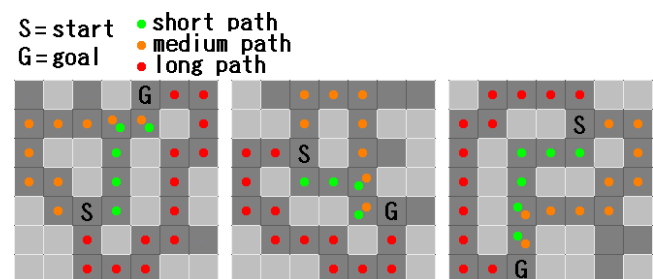


Fig. 2. Randomly generated detour mazes on a 7x7 grid. Dot colours indicate path lengths. Blockage on a cell with only a green dot obstructs only the short path, while blockage on a cell with both a green and an orange dot blocks both the short and medium path.

Many other species (as well as standard reinforcement learning algorithms) are quite capable of learning the shortest path in a maze, but have to re-learn whenever the layout of the maze changes. In this case, we may assume that no representations but simple action-sequences or state-action pairs were learned.

Second Order Learning

Here we place the detour maze task in the theoretical framework introduced above. The maze task is composed of paths (or "accessible space", to be more precise), walls ("inaccessible space"), and a food reward. These are the aspects of e , implementing E . We see that a simple change in e (replacing one piece of accessible space with inaccessible space) calls for complex changes in E and consequently for complex changes in L_0 (running a different path altogether). We also find ourselves strongly inclined to ascribe the ability to mentally represent spatial layouts to an animal if it can make this complex update of L_0 in an instant (without further exploration) upon observing the blockage. We know that when we ourselves update our behaviour in such manner, we do so using our mental representation ability.

We said that mental representation is a form of acquired isomorphism. We saw that our framework explains the evolution of isomorphism-acquisition at L_1 as a consequence of the evolution of L_2 . Can we recognize L_2 in the detour maze experiment?

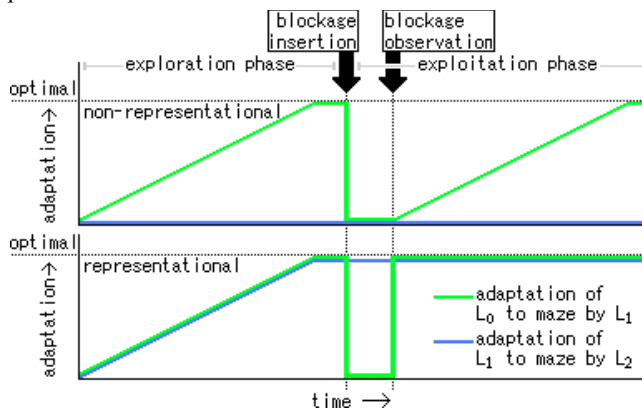


Fig. 3. Adaptation of L_0 and L_1 by L_1 and L_2 in the detour maze task. When a blockage is inserted, behaviour (L_0) inevitable becomes outdated. However, if first order learning (L_1) has been adapted (by L_2) to the maze, then optimality of behaviour can be restored with minimal information (observation of blockage alone).

When after blockage of the shortest path a rat infers the new optimal path without additional exploration, we can view this inference as a split-second L_1 process: a stimulus (observation of the location of the blockage) produced a change in behaviour (the subject abandons the blocked path and switches to the new optimal path). For L_1 to produce such a fast and effective behaviour-update, L_1 itself must have been adapted to the maze (the update cannot be the result of fixed pre-existing learning ability, as the information in the observation alone does not suffice to explain the update without reference to the specific layout of this maze). In other words, an L_2 process

must have optimized L_1 to the current environment: the optimal update in behaviour has come to be causable by minimal information.

L_2 can be said to pre-emptively associate future stimuli with suitable behaviour updates. This would be infeasible for almost all L_1 , but if L_1 employs isomorphism (here: the isomorphism between the cognitive map and the environment), then L_2 becomes quite feasible.

Hopefully it is clear how this is a concrete instance of the effect abstractly hypothesized earlier on. We suggest that equivalent reformulations can be given for many or all other scenarios that we take to indicate MR. We omit detailed examples here, but the general form is as follows: Consider an environmental object X to be represented. We (should) perceive our subject as representing X if and only if it can pre-emptively adjust its behaviour so as to avoid or bring about specific unseen situations involving X after some period of observation of X . In all such cases, observation affects future changes in behaviour. To the extent such pre-emptive adaptation characterizes MR, explanation in terms of second-order learning should be applicable. In the focal case of cognitive maps, X is the maze, but the general scheme may equally well describe a scenario of spontaneous novel tool use (a scenario generally recognized as involving MR), with X being the tool.

We hypothesized that evolution of second order learning causes evolution of mental representation, but we haven't said anything yet about what it takes to evolve second order learning. We know that the neural basis for learning ability is neural plasticity. Would second order learning require second order plasticity? We would need neural circuitry that can not only change its input-output relation in response to stimulation, but also the way the input-output relation changes in response to stimulation. Such second order plasticity can quite simply be achieved by stringing two plasticity loci together on a neural pathway (examples are given below in the next section). So second order learning should be evolvable from standard neural plasticity, but only if we allow for multiple independent plasticity loci to exist along neural paths between input and output neurons².

Given that we said that MR is characterized by second order learning, and that second order learning depends on second order neural plasticity, we see that our hypothesis makes two predictions.

- P1.** In principle, the abilities that characterize mental representation ability can evolve from second order neural plasticity.
- P2.** It is impossible to evolve the abilities that characterize mental representation in a species restricted to first order neural plasticity.

If true, Prediction 1 should be confirmable empirically by taking a suitable artificial species with second order plasticity, evolving it in an environment composed of tasks requiring mental representation, and observing whether it evolves to

² This may sound like a weak requirement, but note that error back-propagation neural networks do not meet it. It follows that such networks are incapable of implementing L_2 , and therefore their L_1 cannot be exposed to selection for isomorphism, making them unsuitable for evolution of representational cognition.

solve those tasks. Note that failure of such a species to evolve MR would not disconfirm our hypothesis: prediction 1 states merely a possibility, not a necessity. Prediction 2, on the other hand, cannot feasibly be confirmed empirically, as we would have to test every possible first order learning species. However, even a single counter-example against prediction 2 would disconfirm our hypothesis, so any computational successes should be analyzed to verify that evolved solutions use at least second order plasticity.

Model

We test the hypothesis using a model in which neural nets with the basic elements for second order neural plasticity are evolved in an environment containing detour mazes.

Environment

The task environment is composed of detour mazes and various simpler maze tasks. An environment composed of detour mazes alone was found ineffective. This is unsurprising: the evaluation criteria of the detour task evaluate MR ability only, i.e. the ability to walk the correct path from a choice of paths after the preferred path has become blocked, but our species starts out unable to walk any path at all (the initial generations spend most of their time bumping into walls helplessly). Inclusion of simpler tasks facilitates evolution of the sub-skills necessary for the detour task. Each task has an exploration phase in which the agent should locate the target, and one or more exploitation phases in which it should run to target in as few steps as possible. The full set of tasks is as follows:

1. An open field. Here there are no walls (aside from the edges of the grid-world). The start position differs between exploration and exploitation phase. This task facilitates evolution of the ability to memorize a location by (geocentric) coordinates (a skill called "place learning" by Tolman, 1948). Exploration time: 200 steps.
2. A "maze" with just a single path from start to finish. Simply following the path leads to reward. This task facilitates evolution of the ability to walk a path. Exploration time: 100 steps.
3. A "dark" version of task 2. Here no visual input (i.e. wall perception) is given during the exploitation phase. This task facilitates evolution of the ability to memorize a sequence of actions (the shape of the path). Exploration time: 100 steps.
4. A two-path maze (one short path, one long path) with dynamic path-blocking. This task has three exploitation phases. In the first, the agent has to pick the short path. In the second, the short path is blocked. The agent is expected to try the short path, find it blocked, then back-track and pick the long path. In the third, the agent should remember that the short path is blocked, and pick the long path straight away. Exploration time: 150 steps.
5. Detour mazes, as described above. Here too there are three exploitation phases, handled just like in task 4, but now with the added difficulty of having to pick the correct path out of the two paths that remain after the short path is blocked. Each agent is evaluated in two detour mazes, one in which the medium path is the correct

choice and one in which the long path is the correct choice (exposing each agent to both gives a more representative fitness signal than when this aspect is randomized. The same could be achieved by exposing each agent to a large number of detour mazes, but this gets computationally expensive. Agents are reset to their innate phenotype between tasks, so no inference about path choice can be made from prior tasks). Exploration time: 200 steps.

Tasks 4 and 5 are further complicated by the presence of arbitrary dead ends (as seen in Figure 1: cells without any coloured dots are dead ends). New mazes are generated continuously over the course of the experiments, to avoid over-fitting to any given maze-set. In tasks 1, 2, 3 and 4, fitness is awarded for proximity to the target at the end of the exploitation phase, by the following fitness function:

$$f = 1 - \left(\frac{d_t}{d_s}\right)^p \quad (1)$$

Where d_t is the distance to the goal at the end of the exploitation phase, d_s the distance from the start to the goal, and p a parameter controlling stringency of the fitness function, set to the experiments discussed here. The detour mazes have more stringent evaluation: only actually reaching the target yields a fitness reward (this prevents asymmetrical fitness reward for erroneously picking the medium path and erroneously picking the long path).

Network species

In the environment described above, a population of 100 neural networks is evolved, using a genetic algorithm with mutation but no crossover. Both connection weights (as well as connection types, see below) and network architecture is evolved. Our network species distinguishes itself from standard neural networks by the use of neural grid structures, neuromodulators, and neurotransmitters. We briefly describe these features here.

Neural Grids. Informed by what's known about the neurology of spatial representation (See Moser et al., 2008, for a review), we let the genotype encode not only single neurons, but also neuron grids. We use square grids of three sizes: 1x1 (single neuron), 3x3, and $W \times W$, where W is the size of the world (7 for our 7x7 world). Given the setup of the model, sizes larger than W offer no additional functionality (i.e. $W \times W$ is functionally equivalent to an infinite grid).

The nets have one 3x3 grid and a number of 1x1 grids receiving input. The 3x3 grid encodes for each of the four cardinal directions whether there is a wall in that direction (on the 4 neurons adjacent to the middle neuron). The 1x1 grids encode whether the current position is the start position, whether the current position is the goal position, and the current phase (exploration or exploitation). Additionally, there are input neurons for bias (always 1.0) and noise (random real numbers from [0,1]). Output is read from two 3x3 grids. From the four neurons corresponding to the cardinal directions, the one with the highest activation is selected, and movement in that direction is performed (if possible). One set is read during

exploration and the other during exploitation (so that the nets can easily evolve specialized behaviour per phase). Connectivity is defined on two levels: inter-grid and intra-grid.

Inter-grid Connectivity. If the genotype defines a connection between two grids, then the phenotype gets uniform connectivity between the neurons in the two grids. If the grids are equal in size, connectivity is one-to-one, otherwise all-to-all. This leads to a highly symmetrical connectivity, which by itself would cause the activation within a grid to remain uniform and redundant. This symmetry is broken by our neurotransmitter logic. We label this neurotransmitter nt-B to distinguish it from our other neurotransmitter (see below).

There are two global nt-B values, nt-Bx and nt-By. These dynamically control (in two dimensions, as the neuron grids are 2D) which connection subsets of an all-to-all projection can transmit activation. When both are zero, then this set comprises connections linking corresponding neurons in the grids, relative to the grid centre (e.g. the centre neuron in the pre-synaptic grid to the centre neuron in the post-synaptic grid, the neuron left of the centre neuron in the pre-synaptic grid to the neuron left of the centre neuron in the post-synaptic grid, etc.). Non-zero nt-B values cause simple offsets, as illustrated in Fig. 4. Currently, nt-Bx and nt-By values are hard-wired to reflect the agent's current x-coordinate and y-coordinate, so signal transfer can shift along with position in space. This makes it relatively easy for evolution to devise nets that store information in different locations in a grid depending on their own position in space: if a smaller grid projects to a larger grid, then the activation pattern on the smaller grid affects only a sub-region of the larger grid. We will call this sub-region the *focal area* of the smaller grid on the larger grid. nt-B does not correspond directly to any biological neurotransmitter, but can be reduced to the species' biologically plausible neurotransmitter (see below) via a trivial network transformation (which, however, increases network size dramatically, so this transformation is not performed).

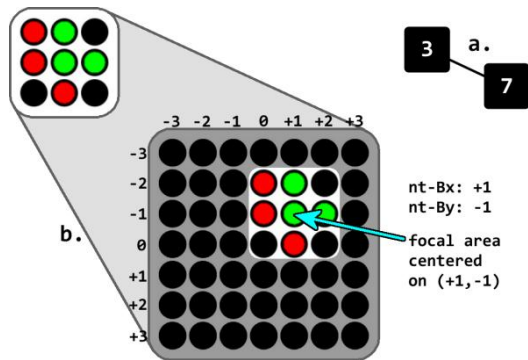


Fig. 4. Neural grids and nt-B. a. Genotype encoding a 3x3 grid, a 7x7 grid, and their connection. b. The corresponding phenotype. The 3x3 grid projects into the focal area of the 7x7 grid. The position of focal areas for projections between unequally sized grids is dynamically controlled by the global neurotransmitter values nt-Bx and nt-By. This mechanism lets the nets conveniently allocate neurons and circuits to specific spatial locations.

Inclusion of coordinates in the input is unnatural, but preliminary experiments focusing on the place learning task (task 1) have shown it quite possible with our model to evolve agents that keep track of their own coordinates without these inputs. Cognitively interpreted, the coordinates in the input and their linkage to the nt-B values make it fairly easy to evolve an innate sense of space as an extended medium in which movement predictably changes one's position. Construction of the ability to represent the volatile and non-uniform *contents* of space, however, is left to evolution.

Intra-grid Connectivity. As the neurons within a grid are not individually represented in the genotype, their connectivity is uniform. Two uniform intra-grid connection patterns are provided: neighbourhood connections (each neuron linking to its four neighbours, with innately identical connections) and reflexive connections (each neuron linking to itself). Neighbourhood connections allow for activation to diffuse over a grid. As neighbourhood connectivity leads to an abundance of loops, linear propagation order cannot be established, so instead we divide each time-step into smaller time-steps in which the activation pattern on grids with neighbourhood connectivity is updated iteratively. Reflexive connections allow for activation patterns to be retained over time (to be precise, a reflexive connection projects from a neuron to its future self, in the next time-step). Reflexive connections are a possible basis for learning, as retention of activation patterns allows acquired activation patterns to influence the behaviour indefinitely. Having multiple reflexive connections in a neural circuit allows for second order learning: if the activation pattern on some grid g_x permanently affects the activation patterns on grid g_y and the activation pattern on g_y permanently affects the formation of the activation patterns on some grid g_z , then g_x has a second order effect on g_z . Such second order effects provide a possible basis for second order learning.

Neuromodulation. Another possible mechanism for both first and second order learning ability is neuromodulation, which we also include in our model. We adopt the variation on the neuromodulation concept from Soltoggio et al., 2008. Neuromodulation provides an evolvable basis for learning ability by making it possible to let networks control their own weight update dynamics. It works as follows: In addition to standard activatory connections, there are modulatory connections. If there is a modulatory connection from neuron X to neuron Y , then activation of X causes modulation of Y . A neuron's modulation value affects the weight updates of its connections. Weights of modulated connections are updated each time-step, using the following update rule:

$$W_{xy} \leftarrow Gr \cdot W_{xy} + A_x^{Gxa} \cdot A_y^{Gya} \cdot M_x^{Gxm} \cdot M_y^{Gym} \quad (2)$$

Where A_x is activation of neuron X and M_x is modulation of neuron X . Gr is a binary gene determining whether the previous value of the weight is included in the update. Gxa , Gya , Gxm and Gym are binary genes controlling for the corresponding pre- and post-synaptic activation and modulation values whether or not they affect connection weight updates. Connection weight values are clipped to the range $[-1, +1]$. Neuromodulation supports second order learning much like reflexive connections do: If there is a modulated connection on the

path from grid g_x to grid g_y , and a modulated connection on the path from grid g_y to grid g_z then g_x can have a second order modulatory effect on g_z .

Of course reflexive connections and modulatory connections can also be combined to form circuits with second order effects. As long as there are at least two points of lasting change on a path from input neurons to output neurons, there is potential for second order changes of the input-output mapping.

Standard Neurotransmitter. The network species has one more special connection type, which transmits a very simple, biologically plausible neurotransmitter, which we label nt-A. As part of the activation function, each neuron multiplies its activation value with its nt-A value. Neurons have an nt-A bias value (genetically defined per group) of 0 (needs to receive positive transmission to be excitable) or 1 (excitable by default, but propagation can be reduced or blocked by negative neurotransmission). Connections of this type only occur in between grids, not within grids. They can be susceptible to nt-B and/or neuromodulation, and have their own set of weight update rule genes.

Results

The model is computationally expensive, and we have insufficient runs with the current version to make definitive statements about its success rates, but so far we have seen a number of successful runs, producing networks with near-optimal performance on all our maze tasks, including the our detour mazes. For our purpose, two aspects of the evolved networks are of particular interest: 1) whether or not they crucially rely on at least second order plasticity circuits, and 2) whether the way the nets solve the maze can be deemed representational in one way or another (i.e. whether the activation or weight patterns acquire any recognizable isomorphism with the maze being explored). We briefly discuss these points for one of the evolved solutions (Fig. 5). The performance of this network in detour mazes is 98% of the theoretical maximum (measured over 4000 maze trials). Observed failures are often the result of incomplete exploration.

Plasticity loci are found at numerous places in the network, but the functionally important ones appear to in the grids marked M2 and M3 in Figure 5. Grid D3 contains a (diffused) copy of the visual input pattern, and forwards this activation pattern via an nt-B controlled projection to M2. M2 uses retention to store the received activation patterns in spatially coherent fashion, forming an image of the maze with positive activation representing accessible positions and zero-activation representing walls. When a blockage is encountered, the negative activation of the D3 neuron for the direction where the blockage is seen knocks out the positive activation on the blockage-position in M2, effectively deleting that position from the image of the maze (the position of the blockage distinguishes itself from other inaccessible positions by its slightly negative value). The image in M2 is used to modify activation flow in M3.

Internally, M3 has positive neighbourhood connections and reflexive connections, however, it has an nt-A bias of zero, meaning that neurons can only activate if they receive positive nt-A from another grid. M2 has a 1:1 nt-A projection

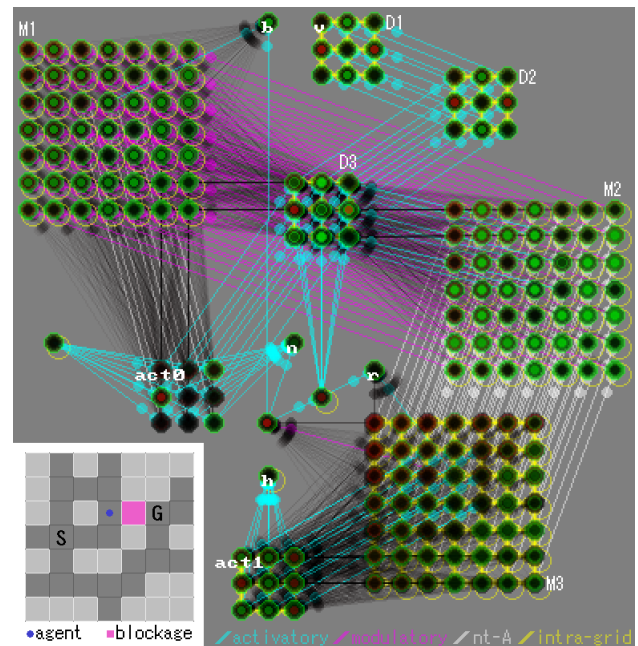


Fig. 5. An evolved solution. Connections run downward.

Functionally irrelevant neurons are removed. D1: visual input. b: bias. n: noise. r: reward. h: home. act0: exploration phase output. act1: exploitation phase output. Grayed-out connections have their transmission blocked on this time-step by nt-B mismatch between their pre- and post-synaptic neurons. Snapshot of network state right after observing the blockage in the exploitation phase of a detour maze task (shown in inset). Activation patterns on grids M1 & M2 can be seen to encode the maze layout, but note that the blockage is only correctly reflected in M2. M3 encodes, at low activation, a gradient over the paths encoded on M2. Output grid act1 reads the activation pattern from its focal area in M3, causing the agent to climb up the gradient during the exploitation phase.

to M3, so the nt-A values on M3 replicate the activation on M2, which in turn replicates the maze layout. The result is that activation diffusion on M3 follows the shape of the maze. Reflexive connections on M3 are innately positive, but sensitive to modulation. At the focal position, modulation is received from a bias neuron, and activation from the reward neuron. The evolved update rule for this connection is absolute (i.e. $Gr = 0$) and takes into account modulation and activation of both the pre- and post-synaptic neuron (though in this case the pre- and post-synaptic neuron coincide, as the modulated connection is reflexive). When the reward neuron is inactive, the result of modulation is that the reflexive connection's weight is set to zero. When the reward neuron is active, positivity of the reflexive connection is retained, and activation inserted at the focal position. The retained positive reflexive connection then ensures that at the neuron at this position retains this activation over time (though it drops off slowly), and the neighbourhood connections let it diffuse over the grid, following the shape of the maze. Note that, as the reward position is the only neuron that retains activation over time, the gradient is effectively recomputed every time-step. Consequently, when the activation pattern on M2 changes (e.g. when a blockage is detected), diffusion flow is instantly rerouted in accordance with the changed maze layout. The

output for exploitation simply reads out the local gradient on its focal area within M3. Optimal choice of path then follows naturally.

What order is this circuit's plasticity, and could it be reduced to 1st order? If we focus on M2 and M3, then we see with three crucial plasticity loci: retention on M2, retention on M3, and modulation on M3. The latter two might be deemed an ambiguous case, with modulation working on reflexive connections. If we consider those a single locus then we are left with two loci. Can we go down to one? No: the functionality of M2 and M3 is not collapsible. For instant adaptation to a layout change, it is crucial that M3 regenerates its activation pattern from scratch every time-step, remembering only the reward position. M2 on the other hand, must hold on to its content over time, because the limits of the net's perception imply that its information can only be gathered in bits and pieces. We conclude that this particular solution relies crucially on second or higher order neural plasticity.

As for the question of whether the solution is representational, we can conclude that the evolved approach clearly employs isomorphism: The maze layout is replicated in the activity patterns of M2 and M3 (as well as M1, although M1's activation pattern does not update in response to observation of a blockage). This solution may be deemed representational.

Different runs of the model produce different solutions. We have for example seen solutions where neuromodulation is used to encode the maze layout in the weights of neighbourhood connections on a 7x7 grid. However, all solutions analyzed so far employ circuits with at least second order plasticity and express the layout of the maze in connection weights and/or activation patterns. We need more successful runs and more extensive analysis before general claims can be made, but these results provide preliminary support for our hypothesis.

Conclusions & Future Work

In this paper we introduced a general hypothesis about how cognitive architectures based on environment-cognition isomorphism may emerge as a consequence of the evolution of learning, and we showed how mental representation ability may be viewed as an instance of this effect. Specifically, we proposed that mental representation may be viewed as the ability for within lifetime acquisition of isomorphism that our hypothesis predicts should evolve under selection for second order learning ability. Given this evolutionary dependence on second order learning, we conjectured that evolution of mental representation requires second order plasticity. We evolved a neural network species that allows for second order plasticity, in an environment containing maze tasks generally believed to require mental representation ability. Successful runs of this model produced network that were found to crucially rely on second or higher order plasticity to solve these mazes, and made clear use of environment-cognition isomorphism, providing preliminary support for our hypothesis.

In this research we clearly used an operational definition of mental representation. We ascribed a species mental representation (in the form of cognitive maps) if it is capable of solving the detour maze task. We should expect the philosophical inclined to take issue with this, so let us state that our choice of definition is purely pragmatic. If it seems behaviouristic, this is only because evolution itself is a behaviourist. Any evolutionary explanation of a mental phenomenon must run via outward behaviour that can be selected on. We haven't touched upon the question of how or why the sort of representation we aim to explain is mental, and we acknowledge this explanatory gap. The objection might be raised that our work then pertains to neural representation only. However, while representation in our evolved networks is clearly neural, we note that our general hypothesis does not make specific claims about the nature of the isomorphism it predicts, requiring merely that it can causally affect behaviour. Depending on how one views the causal powers of mental phenomena, the hypothesis may be equally applicable to the representations we recognize as mental in ourselves.

Beyond improvement of the current model, future directions for our research are extension of this approach to other cognitive domains involving representation, using temporal and social scenarios.

References

- Arnold, S. (2011). Neuro-cognitive Organization as a Side-effect of the Evolution of Learning Ability. *Proceedings of the IEEE Symposium on Artificial Life*, (pp. 100-107).
- Fodor, J., & Pylyshyn, Z. (1988). Connectionism and Cognitive Architecture: a Critical Analysis. *Cognition* (28), 3-71.
- Godfrey-Smith, P. (1996). *Complexity and the Function of Mind in Nature*. Cambridge University Press.
- Minoya, K., & Arita, T. (2011). An artificial life approach for investigating the emergence of a Theory of Mind based on a functional model of the brain. *Proceedings of the 2011 IEEE Symposium on Artificial Life*, (pp. 108-115).
- Moser, E. I., Kropff, E., & Moser, M. (2008). Place Cells, Grid Cells, and the Brain's Spatial Representation System. *Annual Review of Neuroscience* (31), 69-89.
- Soltoggio, A., Bullinaria, J. A., Mattiussi, C., Dürr, P., & Floreano, D. (2008). Evolutionary Advantages of Neuromodulated Plasticity in Dynamic, Reward-based Scenarios. *Proceedings of Artificial Life XI* (pp. 569-576). MIT Press.
- Spencer, H. (1855). *The Principles of Psychology*. New York: Appleton.
- Takano, M., & Arita, T. (2006). Asymmetry between Even and Odd Levels of Recursion in a Theory of Mind. *Proceedings of ALIFE X*, (pp. 405-411).
- Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review*, 55 (4), 189-208.
- Tolman, E. C., & Honzik, C. H. (1930). "Insight" in rats. *University of California Publications in Psychology* (4), 215-232.

On the Relationship Between Environmental and Mechanical Complexity in Evolved Robots

Joshua E. Auerbach¹ and Josh C. Bongard¹

¹Morphology, Evolution & Cognition Laboratory
Department of Computer Science
University of Vermont
Burlington, VT 05405
joshua.auerbach@uvm.edu

Abstract

According to the principles of embodied cognition, intelligent behavior must arise out of the coupled dynamics of an agent's brain, body, and environment. This suggests that the morphological complexity of a robot should scale in relation to the complexity of its task environment. This idea is supported by recent work, which demonstrated that when evolving robot morphologies in simple and complex task environments more complex robot morphologies do tend to evolve in more complex task environments. Here this idea is extended to examining the mechanical complexity of evolved robots. Counter to intuition it is found that the mechanical complexity decreases in more complex task environments.

Introduction

Proponents of embodied cognition posit that intelligent behavior is a product of the coupled dynamics between an agent's brain, body, and environment (Brooks, 1999; Anderson, 2003; Pfeifer and Bongard, 2006; Beer, 2008). Accordingly, the complexity of an agent's brain (control policy) as well as its physical body (morphology) should vary in proportion to the complexity of its task environment. Studying this hypothesis can be approached in several ways. One can investigate the relationship between control and morphology, as was done by Paul (Paul, 2006), and one can also study the relationship between task environment and morphology which is less well understood. In recent work (Auerbach and Bongard, 2012) we began to investigate this latter relationship by studying how the shape complexity of robot body parts varied when robots were evolved in more or less complex task environments. Here, that work is extended by studying a different aspect of morphological complexity: mechanical complexity, a function of the mechanical degrees of freedom of evolved robots.

The experiments presented in this paper fall within the domain of evolutionary robotics (ER) (Harvey et al., 1997; Nolfi and Floreano, 2000). In general ER refers to the practice of employing evolutionary algorithms for the purpose of creating robot control policies and/or morphologies. In the majority of ER studies, control strategies are evolved for human designed or bio-mimicked robot body plans, but it is

also possible to use evolutionary algorithms to create complete robots: placing not only robot control strategies under evolutionary control, but the robots' physical morphologies as well. Evolving morphology, in addition to control policy, allows for the discovery of body plans uniquely suited to a machine's given task environment and presents a systematic way to study the relationship between a robot's morphology and the task environment in which it evolved.

The idea of placing both the morphologies and controllers of robots acting in virtual environments under evolutionary control was first introduced by Sims (Sims, 1994). Sims' work has been followed by subsequent studies (e.g. Lund and Lee (1997); Adamatzky et al. (2000); Mautner and Belew (2000); Lipson and Pollack (2000); Hornby and Pollack (2001a); Komosinski and Rotaru-Varga (2002); Stanley and Miikkulainen (2003); Eggenberger (1997); Bongard and Pfeifer (2001); Bongard (2002); Auerbach and Bongard (2010a, 2011)) which also explored evolving the morphologies and control policies of simulated machines in virtual environments. These studies each had different methodologies and focuses, and the current work differs in a number of important ways.

The most visible ways in which the current study differs from all of these previous studies are (a) how morphological components are modeled and (b) the task environments within which robots evolve. In the majority of previous studies morphologies were built out of interconnected geometric primitives such as cuboids or spheres. These components are easy to model, but severely limit how complex an evolving morphology may become, and therefore restrict what task environments an evolved robot is able to succeed in. This was not a problem for the majority of earlier studies as they commonly restricted themselves to evolving locomotion over flat terrain: maximizing the distance that a robot can displace itself within a given amount of evaluation time. Here, however, more complex task environments are investigated that require the creation of more complex morphologies. Therefore, morphologies should be modeled in a manner which does not have such a low ceiling of complexity. Specifically, in the current work, morphologies are

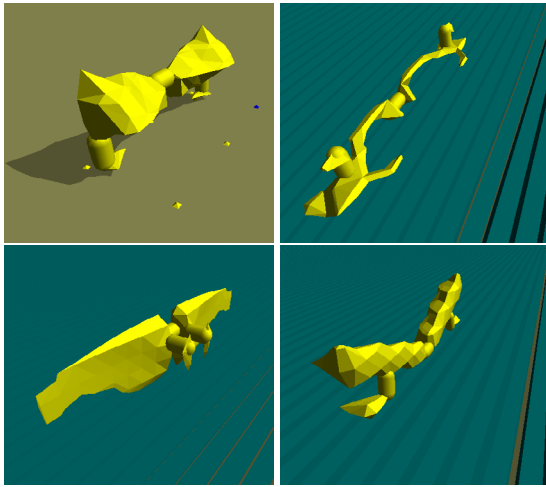


Figure 1: The simple flat ground control task environment (upper left) and three of the experimental task environments with robots that evolved to locomote in each. The ground is a high friction surface, while the blue “blocks of ice” have very low friction. Videos of these robots in action are available online at <http://tinyurl.com/ALife13-Videos>

composed of a number of triangular meshes (trimeshes). Trimeshes can model arbitrary shapes and thus allow for the creation of more complex morphologies than is possible with cuboids or spheres (see Figure 1 for examples).

The current study also differs from much previous work in this domain in the manner by which the robots’ genomes are encoded and evolved. Morphologies in the current work are encoded with Compositional Pattern Producing Network (CPPN) genomes (Stanley, 2007) which are evolved using CPPN-NEAT: an extension of the widely used NeuroEvolution of Augmenting Topologies (NEAT) algorithm (Stanley and Miikkulainen, 2001). CPPNs are a form of indirect encoding inspired by developmental biology possessing many advantages over other encodings (for more details see (Stanley, 2007; Stanley et al., 2009; Clune et al., 2009a,b; Auerbach and Bongard, 2010b,a, 2011)). This is particularly true for robot morphologies as it has been shown previously (Hornby and Pollack, 2001b; Komosinski and Rotaru-Varga, 2002) that generative and developmental encodings have demonstrable benefits over direct encodings in this domain.

Following the methods introduced in (Auerbach and Bongard, 2012), here robots are evolved not only to locomote over flat terrain, but to locomote in a number of more complex, icy task environments as well. However, while in that study robots were restricted to having two mechanical degrees of freedom, here robots are allowed more flexibility in their construction including the ability to utilize a greater number of degrees of freedom. How the robots evolve to

use (or not use) these additional degrees of freedom in different task environments is the main object of study. Here we define mechanical complexity to be the number of mechanical degrees of freedom in an evolved robot. This form of complexity can be considered an aspect of morphological complexity, but as will be shown, mechanical complexity is an orthogonal direction of complexity to the type of morphological complexity discussed in (Auerbach and Bongard, 2012), and provides additional insight into the relationship between task environments and the robots evolved inside them.

The rest of this paper is laid out as follows: first the CPPN encodings are described in more detail including how they evolve and how actuated robots are produced from them. Following this the simulated task environments in which robots are evolved are described including a brief discussion of previous experiments in these task environments and why the particular task environments employed here were chosen. Next, results are presented demonstrating how the mechanical complexity of evolved robots varies across these different task environments with counterintuitive results. This is followed by a discussion of these results and what conclusions may be drawn from them.

Methods

CPPNs

As mentioned in the introduction this study employs Compositional Pattern Producing Networks (CPPNs) for the purpose of encoding populations of evolving robots. CPPNs may be considered a form of artificial neural network (ANN). However, while traditional ANNs are often used as control policies for evolved robots, CPPNs are more often used as genomes for producing some other object of interest. Past work has employed CPPN genomes to evolve pictures (Stanley, 2007), 3D structures (Auerbach and Bongard, 2010b; Clune and Lipson, 2011), robot morphologies (Auerbach and Bongard, 2010a, 2011) or traditional ANNs themselves (Stanley et al., 2009; Clune et al., 2009a; Verbancsics and Stanley, 2011). Here CPPNs are similarly employed to produce actuated robot morphologies.

CPPNs differ from traditional ANNs in several other important ways. While traditional ANNs typically use the same activation function (such as a sigmoid or a step function) at every node, CPPN nodes can take on one of several activation functions from a predefined set. This set typically contains functions that are symmetric such as Gaussian as well as repetitive functions such as sine or cosine. Using functions with these properties allows CPPNs to produce outputs with properties commonly seen in natural systems: symmetry, repetition, and repetition with variation. A more thorough discussion of CPPNs and their properties is beyond the scope of this paper. More details are available elsewhere in the literature (Stanley (2007) for example).

Evolutionary Algorithm

Similar to most other studies employing CPPN genomes, the CPPN-NEAT (Stanley, 2007) evolutionary algorithm is employed to evolve CPPNs in this work. In CPPN-NEAT the state of the art NeuroEvolution of Augmenting Topologies (NEAT) (Stanley and Miikkulainen, 2001) algorithm for neuro-evolution is extended to evolve CPPNs. In this algorithm the CPPNs in the initial population are created to be minimally complex. That is, initially the networks do not have any internal or hidden nodes. Over evolutionary time the complexity of networks in the population is allowed to gradually increase through the creation of additional nodes and links. Often adding additional components to an evolving network will cause the fitness of its phenotype to decrease. NEAT compensates for this by dividing the population into “species” thus allowing novel structural innovations time to mature and promoting genotypic diversity to prevent pre-mature convergence to local optima. For a complete description of how NEAT and CPPN-NEAT work, and further discussion of their beneficial properties, the reader is directed to (Stanley and Miikkulainen, 2001; Stanley, 2007).

Building Robots from CPPNs

Recently (Auerbach and Bongard, 2012) we introduced a system for using CPPNs to create actuated robot morphologies composed of triangular mesh components, which is extended here. This method differs from previous studies (Auerbach and Bongard, 2010a, 2011) where robots were constructed from evolving CPPNs by attaching spherical components to each other by means of an iterated growth procedure. While these earlier studies produced promising results, the methods they employed have several undesirable properties. The extra indirection created by the growth procedure used there prevents many of the desirable features of CPPNs (discussed above) from being realized in the morphologies they produce. Additionally, while it is easy to physically simulate spheres as they have single points of contact, it is possible to create much more complex morphologies using trimeshes.

Trimeshes do require more computational resources to simulate however, as they do not have such simple contact models as spheres, and require the use of smaller simulation step sizes to be stable in the task environments investigated here. However, all experiments described in this paper were carried out on a 7.1 teraflop supercomputing cluster¹, thus making these simulations feasible.

As opposed to employing a growth procedure to create morphologies from CPPNs the current study employs a voxel based method to create morphologies out of trimesh components. This is similar to what is done for the creation of 3D shapes in (Clune and Lipson, 2011). A regular grid is

placed over a region of 3D-space which defines the presence of voxel locations. In the current work this region extends from -1 to 1 (inclusive) in each dimension and grid lines are placed at intervals of 0.2 . This yields a total of 11 grid lines in each dimension for a total of 1331 voxels, this is the same discretization that was applied in (Auerbach and Bongard, 2012).

A candidate CPPN is iteratively queried with the (x, y, z) Cartesian coordinates at every voxel location except for the extrema in each direction. Voxel locations that exceed a pre-defined output threshold (0.5 in this case) are considered to contain matter, while those that do not exceed this threshold are considered to be devoid of matter. All voxels lying on one of the extrema ($|x| = 1$ or $|y| = 1$ or $|z| = 1$) are given output value 0 to ensure that the final triangular meshes have completely enclosed surfaces. Once the CPPN has been queried for every voxel location, the Marching Cubes algorithm (Lorensen and Cline, 1987) is employed to create triangular meshes from the underlying voxel data. Specifically an enclosed triangular mesh is created for each connected voxel component which defines the exterior surface of a single physical shape. These triangular meshes are sent to the physics simulator where they define the exterior surfaces of solid objects and are imbued with mass. As far as the authors are aware prior to (Auerbach and Bongard, 2012) physically simulating evolved, rigid body robots composed of triangular meshes had not been previously reported in the literature.

Our previous work concerned itself with investigating how different task environments affect the shapes of evolved morphologies. To accomplish this goal a single enclosed trimesh component out of the many possibly produced from a CPPN was selected and then reflected and copied in order to form a bilaterally symmetric, two mechanical degree of freedom, actuated robot. Here, however, the primary object of study is the mechanical complexity of the evolved robots, so more components are needed. The current system requires that a candidate CPPN produce at least two enclosed trimesh components. The two largest components A and B are then selected to produce an actuated robot. This is done as follows. First the vertices $a \in V(A)$ and $b \in V(B)$ are found that minimize

$$|\vec{ab}| \quad \forall (a, b) \in V(A) \times V(B)$$

where $V(A), V(B)$ are the vertices of A, B respectively. Next, the component with larger minimum z -coordinate of A, B is translated along \vec{ab} (or \vec{ba}) until it is 0.2 units away from the other component, and the two components are connected together via an intermediary capsule (capped cylinder) of length 0.2 units and radius 0.1 with major axis defined by \vec{ab} . The trimesh components may connect via this intermediary capsule by means of two joints, each being a single degree of freedom rotational (hinge) joint. These joint will have rotation normals determined by \vec{ab} . Specifically

¹The Vermont Advanced Computing Core (VACC),
<http://www.uvm.edu/vacc>

Parameter Name	Symbol	Range of allowed values	Interpretation
Enable Flag	f	[0.0, 1.0]	If $f > 0.5$, then the corresponding joint is enabled, else disabled.
Amplitude	a	[0.25, 0.75]	If the joint is enabled, then it is actuated by an oscillation between $-a\pi$ and $a\pi$ radians, additionally the joints range of motion is restricted to this range.
Period	p	[250.0, 1500.0]	If the joint is enabled, then its oscillation will have a period of p simulation time steps
Phase Shift	s	[-1.0, 1.0]	If the joint is enabled, then its oscillation will be offset from the global oscillation by s periods

Table 1: Description of the four floating point parameters evolved for each of the six potential mechanical degrees of freedom.

two rotation normals that are orthogonal to each other and orthogonal to \vec{ab} are chosen. Since these two joints effectively define a universal joint, the specific normals are unimportant as long as they are orthogonal to each other and to \vec{ab} , so the first \vec{n}_1 is chosen arbitrarily (but consistently) to be orthogonal to \vec{ab} and the second \vec{n}_2 is computed as $\vec{ab} \times \vec{n}_1$.

Once the two trimesh components are connected together with their intermediary capsule the whole object including the connecting joints is reflected across the x -axis as was done with the single trimesh component in (Auerbach and Bongard, 2012). These objects are then spread apart by 0.2 units and once again connected by a capsule of this length. This capsule has its major axis along the y -axis of the coordinate system and connects the two objects at their closest points. These objects each connect to this capsule by means of hinge joints. These joints have rotation normals of (1, 0, 0) and (0, 0, -1) such that the joints rotate through the robot's coronal and sagittal planes respectively. Reflecting and copying the object in this manner ensures that the robots are bilaterally symmetric, which makes locomotion easier, while using two evolved trimesh components instead of the one used in prior work allows for a much greater number of morphologies and locomotion strategies. The two components within each half of the robot may connect in any orientation, and the robots may now have up to six mechanical degrees of freedom.

In addition to the trimesh producing CPPNs, each robot genome possesses a number of additional parameters that are directly encoded as was done in (Auerbach and Bongard, 2012). These parameters are stored as floating point values and are used to determine aspects of the control policy as well as mechanical properties of the evolving robots. Principally, there are six parameters, one for each potential mechanical degree of freedom that act as flags for enabling or disabling a given joint. If a joint is disabled it is replaced with a rigid connection and the remainder of the control parameters relating to that joint are ignored. However, if a joint is enabled it is actuated by means of a coupled oscillator parameterized by its amplitude, period, and phase shift from a global sinusoidal pattern generator. This results in the complete genomes being composed of a CPPN plus a 24-dimensional floating point array (four parameters for each of

the six potential degrees of freedom). These floating point values are recombined and mutated in the same manner as CPPN link weights with mutation magnitudes scaled by the range of values for that parameter. Additionally, crossover on these vectors is possible in all instances of sexual reproduction since every individual contains a vector of the same dimensionality. These parameters, their ranges, and their meanings are detailed in Table 1. Each parameter has a mutation probability of 0.1, same as used in (Auerbach and Bongard, 2012).

Allowing each degree of freedom to be enabled or disabled in this manner allows evolution to adjust the number of mechanical degrees of freedom as necessary and therefore be able to tune the mechanical complexity of the evolved robots. Moreover, encoding the control parameters in this fashion is done to keep the controllers as simple as possible so that fitness is primarily dictated by the morphologies of the robots while at the same time allowing for diverse enough behavior so that the robots can succeed in the different task environments investigated.

Selecting desirable robots

A candidate robot, including two enclosed triangular meshes, joint enable flags, and accompanying control parameters are sent to a physics simulator² and allowed to act for a fixed number of simulation time steps. Similar to (Auerbach and Bongard, 2012) robots are allowed to move for $T = 12500$ time steps. While this is a much greater number of time steps than has been employed in earlier studies (e.g. 2500 in (Auerbach and Bongard, 2011)) it is chosen in order to simulate a comparable amount of real world time. The reason such a large T is necessary is because a very small step size of 0.001s is used in this work. This small step size is necessary to stably simulate the sorts of simulated robots employed here in complex environments.

After the robot has completed its time in the simulator its fitness is calculated. This fitness calculation is exactly the same used in (Auerbach and Bongard, 2012). It is designed to prevent evolution from "cheating" as it often does

²Simulations are conducted in the Open Dynamics Engine (<http://www.ode.org>), a widely used open source, physically realistic, simulation environment.

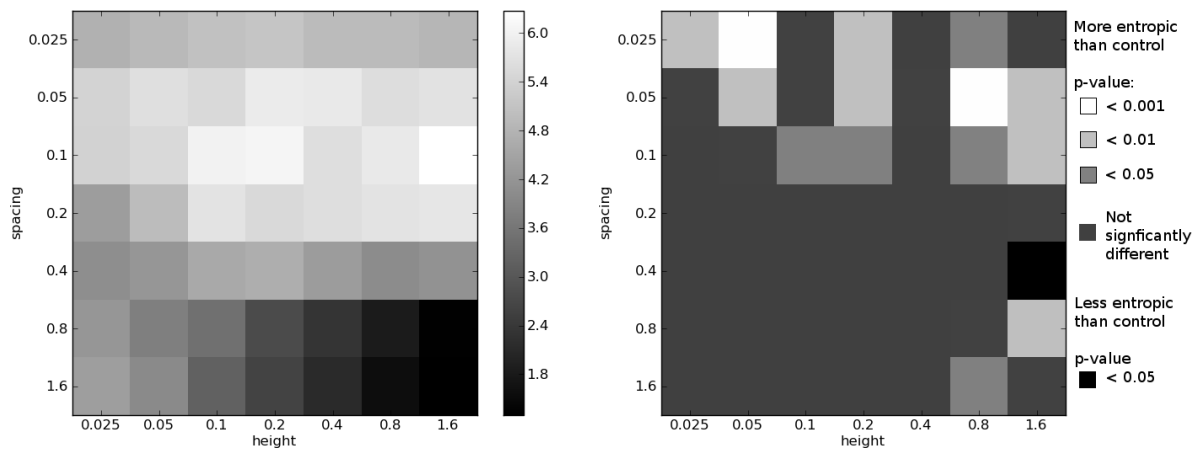


Figure 2: Results from (Auerbach and Bongard, 2012). **(Left)** Mean distance achieved (in arbitrary ODE units) by best individual in final generation taken across 100 independent runs in each of 49 experimental task environments investigated there. For comparison the mean distance achieved from 100 independent runs in a flat ground control task environment was 5.09 units. **(Right)** The ways in which morphologies from experimental environments were more, less, or equally complex (entropic) compared to those evolved in the control task environment. The more complex experimental task environments tended to select for more complex morphologies: there were many experimental task environments where significantly more complex morphologies evolved, while only one experimental task environment selected for significantly less complex morphologies. All p-values were calculated using the Mann-Whitney U test. Figure taken from (Auerbach and Bongard, 2012)

with naïve fitness functions. While a detailed explanation of the ways in which evolution may “cheat” different fitness functions is provided in that paper, here we simply state that fitness is calculated as $\min p(T)_x - \max p(0)_x$ where $\min p(T)_x$ is the minimum x -coordinate of any point on the robot at time T , and $\max p(0)_x$ is the maximum x -coordinate of any point on the robot at the start of the evaluation.

Using this method of fitness evaluation robots are evolved with CPPN-NEAT for 500 generations with a population size of 150 individuals. The implementation of CPPN-NEAT including its parameter settings and CPPN activations functions are the same as employed in (Auerbach and Bongard, 2012).

Choosing task environments

Previously, with the robots composed of a single enclosed trimesh that was reflected and copied, we explored evolving robots in a large number of task environments with the goal of studying how morphological complexity varies in relation to environmental complexity. These task environments consisted of a control environment with flat, high friction ground similar to that used in many other studies, and experimental task environments with an infinite series of low frictions rectangular solids, or “blocks of ice”, fixed in place on top of the ground. These “ice blocks” were constructed such that it was impossible for a robot to gain purchase by moving over their upper surfaces but needed instead to reach into the gaps between the blocks to propel themselves forward. This required the evolution of morphologies with appropri-

ate physical forms. A large number of these icy task environments were explored varying according to two parameters: the height of the blocks and the spacing between the blocks. While the relative complexities of different icy environments were not considered, all the icy environments are considered to be more complex than flat ground because they have greater Kolmogorov Complexity (Kolmogorov, 1965).

Figure 2 revisits these results. It shows, for robots evolved in that work, both how mean fitness varied across task environments and how the evolved robot morphologies differed in complexity when compared to those evolved to locomote in the flat ground, control, environment³. These results are employed here to select task environments for investigation with the current system.

Robots evolved with the current system, employing two trimesh components and three capped cylinders with up to six actuated mechanical degrees of freedom, are slower to simulate than those evolved previously. Due to this slowness, and additional time constraints, it was not possible to experiment with evolving robots in all 50 task environments previously investigated. In lieu of that, robots in the current study are evolved in the flat ground control environment plus five experimental environments. These five experimental environments are chosen based on previous results to be those

³The measure used for comparing morphological complexities, H_Δ , is a measure of shape complexity based on Shannon Entropy (Shannon, 1948) that has been previously shown to correlate with human intuitions of complexity (Page et al., 2003; Sukumar et al., 2008). The reader is referred to (Auerbach and Bongard, 2012) for a description of this measure.

within which robots could be successful and which selected for the most morphologically complex robots (see Figure 2). Specifically the five environments chosen are: blocks of ice 0.8 units tall spaced by 0.05 units (*Environment 1*), blocks of ice 0.05 units tall spaced by 0.025 units (*Environment 2*), blocks of ice 1.6 units tall spaced by 0.1 units (*Environment 3*), blocks of ice 1.6 units tall spaced by 0.05 units (*Environment 4*), and blocks of ice 0.2 units tall spaced by 0.05 units (*Environment 5*). These five task environments cover a variety of these parameters and should be a good sampling of the overall parameter space.

Results

For each of the six task environments investigated: the control plus five experimental task environments, 50 independent experimental runs of CPPN-NEAT were conducted⁴. As can be seen in Figure 3, in each environment studied this system is capable of evolving robots that successfully locomote in the desired direction. Though, due to using the same number of evaluations in an enlarged search space the robots produced in the final generations here tend not to locomote as far as those evolved previously (compare to the left of Figure 2). However, the absolute performance of these robots is not of primary interest in this paper.

Of greater concern is how the mechanical complexity of the evolved robots varies from the simple control environment to the more complex experimental task environments. Towards this aim Figure 4 plots the mean number of mechanical degrees of freedom that robots evolved to use in each task environment. Counter to intuition the simple task environment actually selects for more mechanically complex robots: the robots evolved in the simple task environment have significantly more mechanical degrees of freedom on average, than those evolved in each of the five complex task environments. This is corroborated by Figure 5 which shows that the flat ground task environment not only selects for a greater number of mechanical degrees of freedom but that the degrees of freedom that are selected for have a significantly greater range of motion on average than the degrees of freedom in robots evolved in each of the more complex experimental task environments.

Discussion

Why is it that the same task environments which have been shown to select for greater complexity of morphological components select for reduced mechanical complexity? Intuitively these two forms of complexity should be correlated, but this is clearly not the case here. One hypothesis is that the reduction of mechanical complexity in the icy task environments is due to them being more difficult than the flat

⁴While 50 runs were started for each task environment, a small number of runs failed to complete for each of the experimental task environments. The results reported here only include those runs that completed successfully.

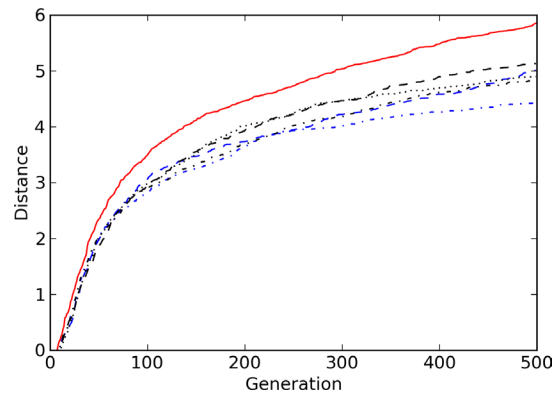


Figure 3: Mean distances by generation achieved by robots evolved in the control environment (red) and each of the five experimental task environments (*env. 1* blue dashes, *env. 2* blue dash-dots, *env. 3* black dashes, *env. 4* black dash-dots, *env. 5* black dots).

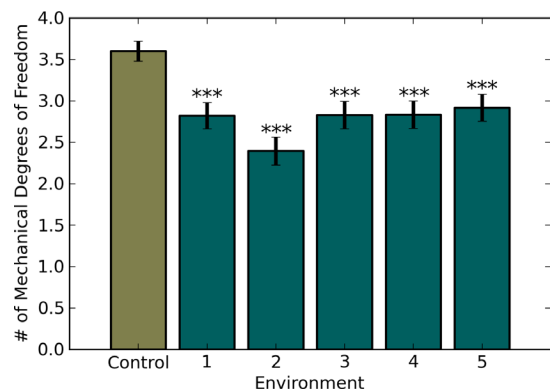


Figure 4: Mean number of mechanical degrees of freedom with standard errors for robots evolved in each task environment. Robots evolved in each of the icy task environments have significantly fewer mechanical degrees of freedom than those evolved in the control environment, p -values < 0.001 in all cases (Mann-Whitney U test).

ground task environment. As can be seen in Figure 3 robots are not able to evolve to locomote as far in the icy task environments as they are on flat ground. This suggests there may be fewer ways to succeed in the icy task environments, and if it is easier to succeed with less mechanical complexity than there will be selection pressure in that direction. Meanwhile, if flat ground is an easier task environment regardless of mechanical complexity there will be little selection pressure on the number of degrees of freedom of the robots evolved there. However, if this is the case, one would expect each degree of freedom of robots evolved on flat ground to be enabled or disabled with equal probability. But, from looking at Figure 4 it can be seen that this is clearly not the case. Robots evolved in the flat ground task environment

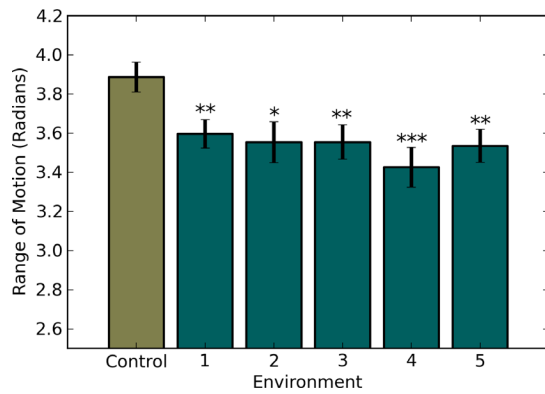


Figure 5: Mean range of motion in radians taken across each enabled joint (mechanical degree of freedom) with standard errors. Robots evolved in each of the icy task environments have significantly smaller ranges of motion than those evolved in the control environment. * denotes p -values < 0.05 , ** denotes p -values < 0.01 , and *** denotes p -values < 0.001 (Mann-Whitney U test).

have a significantly greater number of degrees of freedom than the three that would be expected by equal probability.

Another hypothesis is that there is simply an advantage to having less mechanical complexity in the icy task environments investigated. Succeeding in these environments involves reaching into the gaps between blocks in order to gain purchase, and then coming out of the gaps in order to move forward. Since the robots evolved in this work are all driven by open loop controllers, they have no way of sensing when they are in the gaps or not. It may be that extra mechanical degrees of freedom make it more difficult for the robot to get out of its own way as it traverses the environment. In other words extra mechanical degrees of freedom driven by a sinusoidal control signal cause the robot to often catch itself in the gaps when it could be gliding forward. This seems likely to be the case. As can be seen in the video available at <http://tinyurl.com/alife13-1DOF> it is possible for robots to succeed in these task environments with only a single mechanical degree of freedom and the proper physical shape. This robot only has one actuated joint rotating horizontally but due to its shape it is able to fall into the gaps, gain purchase and glide out of them. Several such single degree of freedom robots evolved in the icy task environments, but only one such robot evolved in the control task environment and it has substantially lower fitness.

While it is counter-intuitive that task environments that select for more complex body components select for less mechanical complexity it makes sense in this instance. It is likely, however that other task environments that are complex in different ways will select for robots that have complex body components and are more mechanically complex. For instance if there existed other obstacles in the environ-

ment that the robot needs to step over one could imagine how additional degrees of freedom would be useful in order to reach over the obstacles in order to gain purchase on their far sides in ways that would not be possible without additional degrees of freedom. Likewise if the spacing between blocks was uneven then most likely the open loop control policies employed here would be unable to succeed. If sensors and closed loop control were employed it may be advantageous to have extra degrees of freedom in order to actively sense the environment and decide how to move.

Conclusion

This work has investigated the relationship between environmental and mechanical complexity in evolved robots. Results of previous work were used to select task environments in which successful, morphologically complex, robots were previously evolved. However, counter to intuition, the robots evolved here were less mechanically complex than those evolved in a simpler control task environment. This demonstrates that these different forms of morphological complexity do not necessarily correlate with each other, but are likely orthogonal.

Moving forward it will be interesting to explore evolving robots in other task environments that are complex in different ways. It is likely that while the task environments investigated here do not select for greater mechanical complexity there exist task environments in which both greater mechanical complexity and greater complexity of body shape will be selected for. Additionally it will be of interest how control complexity varies in relation to these morphological complexity measures. To this aim the current evolutionary system will be extended to allow for more sophisticated closed loop neural network controllers. Are the task environments that select for greater morphological complexity in one way or another also those that select for greater control complexity? Or are these different forms of complexity—morphological, mechanical, and control—independent?

Acknowledgements

This work was supported by National Science Foundation Grant PECASE-0953837 and DARPA M3 grant W911NF-1-11-0076.

The authors also acknowledge the Vermont Advanced Computing Core which is supported by NASA (NNX 06AC88G), at the University of Vermont for providing High Performance Computing resources that have contributed to the research results reported within this paper.

References

- Adamatzky, A., Komosinski, M., and Ulatowski, S. (2000). Software review: Framsticks. *Kybernetes: The International Journal of Systems & Cybernetics*, 29(9/10):1344–1351.
- Anderson, M. (2003). Embodied Cognition: A field guide. *Artificial Intelligence*, 149(1):91–130.

- Auerbach, J. E. and Bongard, J. C. (2010a). Dynamic Resolution in the Co-Evolution of Morphology and Control. In *Artificial Life XII: Proceedings of the Twelfth International Conference on the Simulation and Synthesis of Living Systems*.
- Auerbach, J. E. and Bongard, J. C. (2010b). Evolving CPPNs to Grow Three-Dimensional Physical Structures. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*.
- Auerbach, J. E. and Bongard, J. C. (2011). Evolving Complete Robots with CPPN-NEAT: The Utility of Recurrent Connections. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*.
- Auerbach, J. E. and Bongard, J. C. (2012). On the Relationship Between Environmental and Morphological Complexity in Evolved Robots. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. To Appear.
- Beer, R. D. (2008). The dynamics of brain-body-environment systems: A status report. In Calvo, P. and Gomila, A., editors, *Handbook of Cognitive Science: An Embodied Approach*, pages 99–120. Elsevier.
- Bongard, J. and Pfeifer, R. (2001). Repeated structure and dissociation of genotypic and phenotypic complexity in Artificial Ontogeny. *Proceedings of The Genetic and Evolutionary Computation Conference (GECCO 2001)*, pages 829–836.
- Bongard, J. C. (2002). Evolving modular genetic regulatory networks. In *Proceedings of The IEEE 2002 Congress on Evolutionary Computation (CEC2002)*, pages 1872–1877.
- Brooks, R. (1999). *Cambrian intelligence*. MIT Press Cambridge, Mass.
- Clune, J., Beckmann, B., Ofria, C., and Pennock, R. (2009a). Evolving Coordinated Quadruped Gaits with the HyperNEAT Generative Encoding. In *Proceedings of the IEEE Congress on Evolutionary Computing*, pages 2764–2771.
- Clune, J. and Lipson, H. (2011). Evolving 3d objects with a generative encoding inspired by developmental biology. In *Proceedings of the Eleventh European Conference on Artificial Life (ECAL)*, pages 144–148.
- Clune, J., Pennock, R. T., and Ofria, C. (2009b). The sensitivity of hyperneat to different geometric representations of a problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*.
- Eggenberger, P. (1997). Evolving morphologies of simulated 3D organisms based on differential gene expression. *Procs. of the Fourth European Conf. on Artificial Life*, pages 205–213.
- Harvey, I., Husbands, P., Cliff, D., Thompson, A., and Jakobi, N. (1997). Evolutionary robotics: the sussex approach. *Robotics and Autonomous Systems*, 20:205–224.
- Hornby, G. and Pollack, J. (2001a). Body-brain co-evolution using l-systems as a generative encoding. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 868–875.
- Hornby, G. and Pollack, J. (2001b). Evolving L-systems to generate virtual creatures. *Computers & Graphics*, 25(6):1041–1048.
- Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1):1–7.
- Komosinski, M. and Rotaru-Varga, A. (2002). Comparison of different genotype encodings for simulated three-dimensional agents. *Artif. Life*, 7(4):395–418.
- Lipson, H. and Pollack, J. B. (2000). Automatic design and manufacture of artificial lifeforms. *Nature*, 406:974–978.
- Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21:163–169.
- Lund, H. H. and Lee, J. W. P. (1997). Evolving robot morphology. *IEEE International Conference on Evolutionary Computation*, pages 197–202.
- Mautner, C. and Belew, R. (2000). Evolving robot morphology and control. *Artificial Life and Robotics*, 4(3):130–136.
- Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology*. MIT Press, Cambridge, MA, USA.
- Page, D., Koschan, A., Sukumar, S., Roui-Abidi, B., and Abidi, M. (2003). Shape analysis algorithm based on information theory. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 1, pages I – 229–32 vol.1.
- Paul, C. (2006). Morphological computation: A basis for the analysis of morphology and control requirements. *Robotics and Autonomous Systems*, 54(8):619–630.
- Pfeifer, R. and Bongard, J. (2006). *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press.
- Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, 27.
- Sims, K. (1994). Evolving 3D morphology and behaviour by competition. *Artificial Life IV*, pages 28–39.
- Stanley, K., D’Ambrosio, D., and Gauci, J. (2009). A Hypercube-Based encoding for evolving Large-Scale neural networks. *Artificial Life*, 15(2):185–212.
- Stanley, K. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.
- Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162.
- Stanley, K. O. and Miikkulainen, R. (2001). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:2002.
- Sukumar, S., Page, D., Koschan, A., and Abidi, M. (2008). Towards understanding what makes 3d objects appear simple or complex. In *IEEE Conference on Computer Vision and Pattern Recognition CVPR 2008, Sixth IEEE Workshop on Perceptual Organization in Computer Vision (POCV)*.
- Verbancsics, P. and Stanley, K. O. (2011). Constraining Connectivity to Encourage Modularity in HyperNEAT. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*.

Testing the Variability Selection Hypothesis: The Adoption of Social Learning in Increasingly Variable Environments

James M. Borg¹ and Alastair Channon¹

¹Research Institute for the Environment, Physical Sciences and Applied Mathematics, Keele University, ST5 5BG, UK
{j.borg,a.d.channon}@epsam.keele.ac.uk

Abstract

The variability selection hypothesis predicts the adoption of *versatile* behaviors and survival strategies, in response to increasingly variable environments. In hominin evolution the most apparent adaptation for versatility is the adoption of social learning. The hypothesis that social learning will be adopted over other learning strategies, such as individual learning, when individuals are faced with increasingly variable environments is tested here using a genetic algorithm with steady state selection and constant population size. Individuals, constituted of binary string genotypes and phenotypes, are evaluated on their ability to match a target binary string, nominally known as the environment, with success being measured by the Hamming distance between the phenotype and environment. The state of any given locus in the environment is determined by a sine wave, the frequency of which increases as the simulation progresses thus providing increasing environmental variability. Populations exhibiting combinations of genetic evolution, individual learning and social learning are tested, with the learning rates of both individual and social learning allowed to evolve. We show that increasingly variable environments are sufficient but not necessary to provide an evolutionary advantage to those populations exhibiting the extra-genetic learning strategies, with social learning being favored over individual learning when populations are allowed to explore both strategies simultaneously. We also introduce a more biologically realistic model that allows for population collapse, and show that here the prior adoption of individual learning is a prerequisite for the successful adoption of social learning in increasingly variable environments.

Introduction

It is now widely accepted that the species *Homo sapiens*, to which all modern humans belong, evolved in Africa before leaving to populate the rest of world (Tattersall, 2009). In order to successfully populate new and challenging environments hominins must have developed versatile and robust behaviors and survival strategies, with the most apparent hominin adaptation for versatility being the adoption of extra-genetic learning strategies such as social learning (Tomasello, 1999). This leads us to ask what was it about the environments in which hominins evolved that enabled

them to adapt to be so versatile and ultimately so successful when moving into new and unfamiliar environments. In response to this question numerous authors have suggested a variety of theories and hypotheses regarding the relationship between hominin evolution and the environment (Potts, 1998a). In this work we do not seek to answer the question of how hominins became such expert social learners, we instead test one of the most prominent theories of hominin evolution and versatility, the Variability Selection Hypothesis (Potts, 1996, 1998a,b), using an artificial life simulation.

The Variability Selection Hypothesis

The variability selection hypothesis, as proposed by Richard Potts (Potts, 1996, 1998a,b), predicts the adoption of versatile behaviors and survival strategies, in response to increasingly variable environments. Over the past seven million years there have been a number of what Potts describes as “large disparities” in environmental conditions and a trend toward increasing climatic variation in and around known early hominin locations in eastern and southern Africa, such as the Turkana and Olduvai basins (Potts, 1998a). Evidence for such inter- and intra-generational changes have been found in a variety of climatic indicators including marine oxygen isotope levels (Potts, 1998a,b), providing insight into temperature changes, and ocean dust records (Potts, 1998a), providing evidence for dust plumes arising from strong seasonal rainfalls and prevailing wind patterns. Both of these indicators demonstrate an upward trend in environmental variability during the last seven million years in Africa, and around the world in general. Evidence from these, and other climatic indicators, shows that major shifts in the African climate correlate well with important early technological milestones and speciation events in hominin evolutionary history (Grove, 2011). Key hominin and hominid adaptations such as early bipedality and complex social behavior emerged during these periods of more pronounced environmental variability (Potts, 1998b). Though the climatic evidence for the variability selection hypothesis is impressive, the hypothesis has had very little theoretical work applied to it. Following the call from Potts (1998b) for

a mathematical framework to explore the variability selection hypothesis, and the work of Grove (2011) to that end, we here test the claim that increasing environmental variability is a sufficient selection pressure to elicit the adoption of social learning, in an artificial life simulation.

Social Learning

Social learning is not restricted to humans and their ancestors: it is a widely observed natural phenomenon, with many species using a variety of social learning mechanisms such as imitation, emulation, teaching and the use of public information to produce adaptive behaviors in dynamic and challenging environments (Laland, 2004; Reader and Biro, 2010; Whiten and van Schaik, 2007). It has been suggested that social learning enables animals to better track their environment by assimilating extra-genetic information from others during their lifetimes while avoiding potentially costly individual learning (Boyd and Richerson, 1995).

The effects and benefits of learning have been studied widely in simulation. According to Nolfi and Floreano (1999) learning may be seen as having several adaptive functions within an evolutionary perspective. These include allowing individuals to adapt to environmental change, enabling evolution to use information extracted from the environment, and guiding evolution. Famously Hinton and Nowlan (1987) demonstrated that by using individual learning, populations are able to solve “needle in a haystack” problems due to learning guiding evolutionary search. Best (1999) extended the work of Hinton and Nowlan (1987) by demonstrating that, given the same “needle in a haystack” problem, social learning outperforms individual learning. Further work using simulated robots (Acerbi and Nolfi, 2007), animats (Borg et al., 2011), autonomous robots (Acerbi et al., 2007), ungrounded neural networks (Curran and O’Riordan, 2007), and binary strings (Jones and Blackwell, 2011) has contributed further to our understanding of the evolutionary advantages provided by social learning.

Social Learning in Increasingly Variable Environments

Numerous models and simulations have demonstrated the adaptive advantages, and highlighted potential failings, of learning strategies in environments exhibiting some level of consistent variation (Borg et al., 2011; Boyd and Richerson, 1983, 1995; Grove, 2011; Jones and Blackwell, 2011; Whitehead and Richerson, 2009). In this work we test the hypothesis that increasing, rather than simply consistent, environmental variability is sufficient to elicit the adoption of social learning. To test this hypothesis populations of individuals, constituted of binary string genotypes and phenotypes, are evaluated on their ability to match a target binary string, nominally known as the environment, with success measured by the Hamming distance between the phenotype and environment. Three classes of environment are used.

1. Static environments in which an environment’s target string remains unchanged.
2. Consistently variable environments in which each locus of an environment’s target string switches on or off at regular, frequent, intervals.
3. Increasingly variable environments in which the frequency of change increases over the period of evolution.

For each class of environment, populations exhibiting combinations of genetic evolution, individual learning and social learning are evaluated, with the learning rates of both individual and social learning allowed to evolve. Mean population fitness is recorded for each combination of environment and learning strategy, with data also collected on the evolved rates of social and individual learning and the reproductive fitness of individuals exhibiting different learning rates when both extra-genetic learning strategies are combined.

Our expectations were as follows.

1. Social and individual learning strategies, both separately and in combination, will outperform genetic evolution on all environments.
2. When evolved simultaneously social learning will be favored over individual learning, with individuals exhibiting higher levels of social learning having a higher reproductive fitness, thus showing that social learning is adopted over individual learning in increasing and consistently variable environments.

The Model

The model used is a genetic algorithm with steady state selection, in which individuals, constituted of binary string genotypes and phenotypes of length L , are assessed on their ability to match a binary target string or, as we shall refer to it here, an environment denoted as E (also of length L). A phenotype is assessed by measuring the Hamming distance between it and the environment. A phenotype is initially a copy of the genotype but can acquire information through evolution and learning, which is discussed in more detail later. This may be achieved by one of four strategies.

1. Genetic Evolution - at reproduction random mutations occur with probability p_{mut} at each locus.
2. Individual Learning - at each epoch (iteration of the steady state genetic algorithm) every individual flips each of the bits in its phenotype with probability p_{ind} .
3. Social Learning - at each epoch every individual copies each locus from a random other individual’s phenotype with probability p_{soc} .
4. Individual and Social Learning (Combined) - at each epoch every individual engages in either individual learning or social learning, with equal probability, at each locus in the phenotype.

The learning rate (per locus probability of flipping or copying) is allowed to evolve independently for each individual. That is to say that a population wide learning rate is not set. Both p_{ind} and p_{soc} are floating point values bounded within the range $[0, 1]$.

Variable Environments

Populations are tested on one of the three environmental setups introduced earlier, two of which exhibit some level of variability. Variability is dictated by a sine wave. At initialization each locus l in the environment is assigned a random value f , which is used to determine the binary value of the environmental locus at each epoch (1).

$$E^l = \sin((f^l \times epoch) \times (\pi/180)) \begin{cases} < 0 \rightarrow 0 \\ > 0 \rightarrow 1 \end{cases} \quad (1)$$

The range of values f may be initially set to is determined by which environment the population is being tested on:

1. No Variability (static): $f = 0$
2. Consistent Variability: $f \in N(1.8, \frac{1.8^2}{2})$
3. Increasing Variability: $f \in N(0.018, \frac{0.018^2}{2})$

Values of $f \approx 1.8$ equate to approximately one change per 100 epochs, with 100 epochs being considered to be one generation of the algorithm (where $L = 100$). A value of $f \approx 0.018$ equates to approximately one change per 10000 epochs, or one hundred generations. One change per generation is referred to as *high* frequency variability, one change per ten generations as *medium* frequency, and one change per one hundred generations as *low* frequency. As each environmental locus has a unique initial value of f , the sine wave dictating the value at each locus will be different, thus avoiding uniform environmental change.

For increasing variability tests the f values increase over time. The f value for any environmental locus (E^l) during increasing tests is determined by the initial f value at that locus (f^0), the maximum f value ($f^{max} = 1.8$), the current epoch and the number of epochs the evaluation is permitted to run for (2).

$$f^{epoch} = f^0 + (f^{max} - f^0) \times (\frac{epoch}{epoch^{max}}) \quad (2)$$

Evolution and Learning

Each test is populated by N individuals, each constituted of the following:

- $g \in \{0, 1\}^L$ - *genotype*, an L -bit string
- $h \in \{0, 1\}^L$ - *phenotype*, an L -bit string initially equal to g but subject to learning. The individual's fitness is L minus the Hamming distance between h and E .

- $p_{ind} \in [0, 1]$ - *individual learning rate*, set initially to 0. In populations allowed to learn in this manner p_{ind} may evolve via mutation.
- $p_{soc} \in [0, 1]$ - *social learning rate*, set initially to 0. In populations allowed to learn in this manner p_{soc} may evolve via mutation.

These properties are broadly consistent with the properties used by Jones and Blackwell (2011). However, unlike Jones and Blackwell (2011) the learning rates are not normalized to sum to unity, instead each rate may evolve to a maximum value of 1.

At each epoch two individuals are selected at random from the population for tournament selection. Reproduction then takes place between the tournament winning individual (the one with the higher fitness) and a random individual from the population, the progeny of this reproduction replacing the tournament loser. Reproduction consists of both recombination and mutation. Recombination is by way of single point crossover, where a random position $l \in [0, L - 1]$ is selected. Bits 0 to l being taken from one of the parents and bits $l + 1$ to $L - 1$ from the other, with the order of the parents determined at random at each reproduction. Mutation occurs at each locus in the child's genotype, with probability $p_{mut} = 1/L$ of the bit at that locus being flipped. Following reproduction g is copied without error to h which from this point in the child individual's lifetime is used for fitness evaluation and learning. In learning populations parental values of p_{ind} and p_{soc} are also inherited (depending on the learning strategy implemented for the population). The child inherits one of its parents' learning rates at random, with the learning rate then being mutated by the addition of Gaussian random noise (mean 0, standard deviation 0.01).

Learning comes in two distinct strategies: individual and social. At each epoch all individuals from a learning population are afforded the opportunity to learn. Individual learning takes the same form as mutation at reproduction, with each locus in h bit-flipping with probability p_{ind} . Social learning on the other hand is a little more involved: for each locus in h there is a probability p_{soc} of copying the tournament winning individual's equivalent locus. Copying the tournament winning individual in social learning strategies may be seen as akin to the "*copy-successful-individuals*" strategy outlined by Laland (2004) and implemented (though in a slightly different manner) by Jones and Blackwell (2011). In those populations exhibiting both individual and social learning in combination, which of the two learning strategies to use is chosen at random (50:50) for each locus of each individual, and applied with the appropriate learning rate. Individuals are also afforded the opportunity to unlearn any learned information. Each individual maintains a copy of their phenotype from before learning; if after learning their fitness is less than it was during the last epoch, their previous phenotype is restored.

Experimentation and Results

Experimentation was initially conducted on the static, consistently variable and increasingly variable environments. Forty initially random populations of size $N = 100$ were tested for each environmental setup: ten populations per learning strategy. Each environment, of size $L = 100$, was initially identical in its binary composition, as was the random number seed from which the initial f values were derived. Each population was run for 100000 epochs (1000 generations), with the population being sampled every 100th epoch (once per generation). The data presented here takes the mean performance of each of the ten populations per learning strategy at every generation.

A set of further tests were also conducted to assess in which conditions of environmental variability populations were likely to collapse. These tests were conducted in two differing setups. In both setups N was maintained at 100 but before standard tournament selection took place all individuals with a fitness less than $L/2$ were killed, these individuals being deemed to be unfit. If at this point the new population size $N' \leq N \times 0.1$ the population is considered to have collapsed and evolution is terminated. If the population does not collapse, tournament selection takes place to replace one surviving individual, and the population is then re-populated to $N = 100$ by the progeny of randomly selected other surviving individuals. The first test setup was conducted for a maximum of 100000 epochs, with populations reaching this epoch being considered as surviving populations.

The second population collapse test setup differs from the first in three distinct ways: tests were simulated for 200000 epochs; only populations exhibiting the individual and social learning strategies combined were tested; and social learning was prohibited from being used or evolving for the first half of each experiment.

Static Environments

As can be seen from figure 1(a), under static conditions both social learning and individual and social learning combined perform much better than genetic evolution and individual learning. These results are broadly consistent with those of Jones and Blackwell (2011) who also found social explorations to be advantageous and individual learning to sub-optimal in static environments. However, unlike Jones and Blackwell (2011), in these tests individual learning does not outperform no-learning (genetic evolution alone) over the entire simulation. This result is a little surprising given Hinton and Nowlan (1987), which demonstrates that individual learning should be able to better guide evolution than random mutation alone. It also seems that individual learning is not highly expressed when used in isolation. Figure 4 shows that under unchanging environmental conditions individual learning does not achieve a maximum p_{ind} of above 0.2, this value being lower than in all other environmental conditions and significantly lower than p_{soc} , which in static

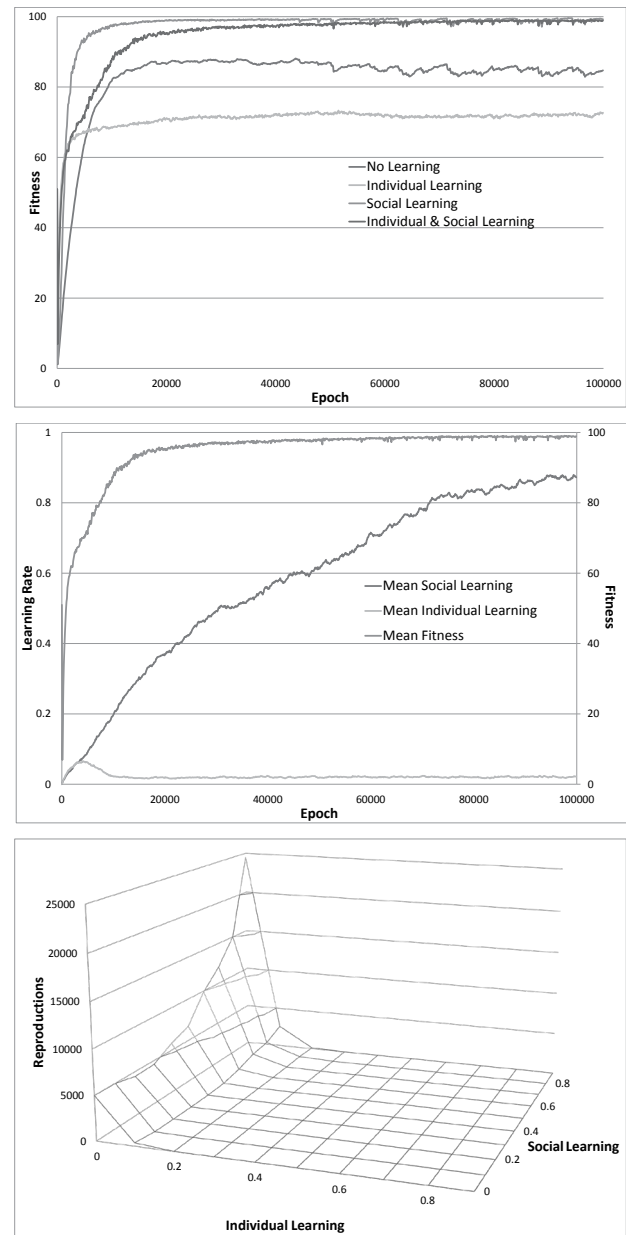


Figure 1: Static Environment Tests: (a) Mean fitness of each learning strategy, (b) Mean fitness of individual and social learning with the evolved learning rates, (c) Reproductive fitness of combined learning rates.

environments achieves a value in excess of 0.7. Individual learning is also marginalized when expressed in combination with social learning. Figure 1(b) shows that when evolved together social learning outstrips individual learning by some distance, with individual learning becoming almost unused after an initial spike before 1000 epochs. Interestingly, for static environments the maximum value of p_{soc} achieved is larger when individual and social learning are found together, than when social learning is evolved in isola-

tion, implying that social learning requires individual learning to be fully expressed. As hypothesized social learning is adopted over individual learning, this adoption also being reflected by the reproductive fitness of individuals exhibiting the combined learning strategy as shown in figure 1(c). Individuals exhibiting intermediate values for p_{soc} and low values (below 0.1) of p_{ind} are shown to be more reproductively fit by contributing to a larger number of reproductions over the evaluation period.

Consistently Variable Environments

As shown in figure 2(a), under consistently variable conditions, where f is maintained at 1.8, the extra-genetic learning strategies all outperform no-learning (genetic evolution alone). In high variability environments non-learners find it difficult to track changes in the environment using mutation and recombination alone, causing populations of non-learners to average out at a fitness of $L/2$: no better than random. Of the extra-genetic learning strategies the combined strategy far outperforms individual and social learning alone. Individual learning when exhibited in isolation tends to find a stable value very quickly, but is unable to improve upon it. Social learning on the other hand rapidly (though also rather noisily) finds highly optimal solutions. However, the ever increasing reliance on social learning, as demonstrated by a maximum learning rate of above 0.9 (see figure 4), causes social learners' fitness to decrease to a value equal to that of individual learners, suggesting that overly conformist learning strategies are no better than trial-and-error personal innovations at tracking high levels of environmental change. By combining individual and social learning the negative aspects of both strategies in isolation seem to vanish: fitness does not stabilize at a sub-optimal value early on and fitness does not decrease over time. This suggests that the conformist bias imposed by social learning is in some way tempered by non-social innovation. However, as we can see in figure 1(b and c) social learning is largely adopted over individual learning, with p_{ind} being sidelined to values well below 0.1 and highly reproductive individuals exhibiting high levels of social learning and low levels of individual learning. The initial spike in individual learning seen early in the combined strategy, while p_{soc} is also low, may indicate that the vast majority of innovation is introduced into the population before it becomes overly conformist. It is also interesting to note that the spike in p_{ind} correlates well with the noisiest fitness period. Once enough innovation is introduced into the population innovation appears to be sidelined, although maintained at a low level, and individuals become increasingly reliant on social learning.

Environments of Increasing Variability

Unlike in consistently noisy environments, all populations exhibiting extra-genetic learning strategies find it difficult to maintain high levels of fitness when confronted with increas-

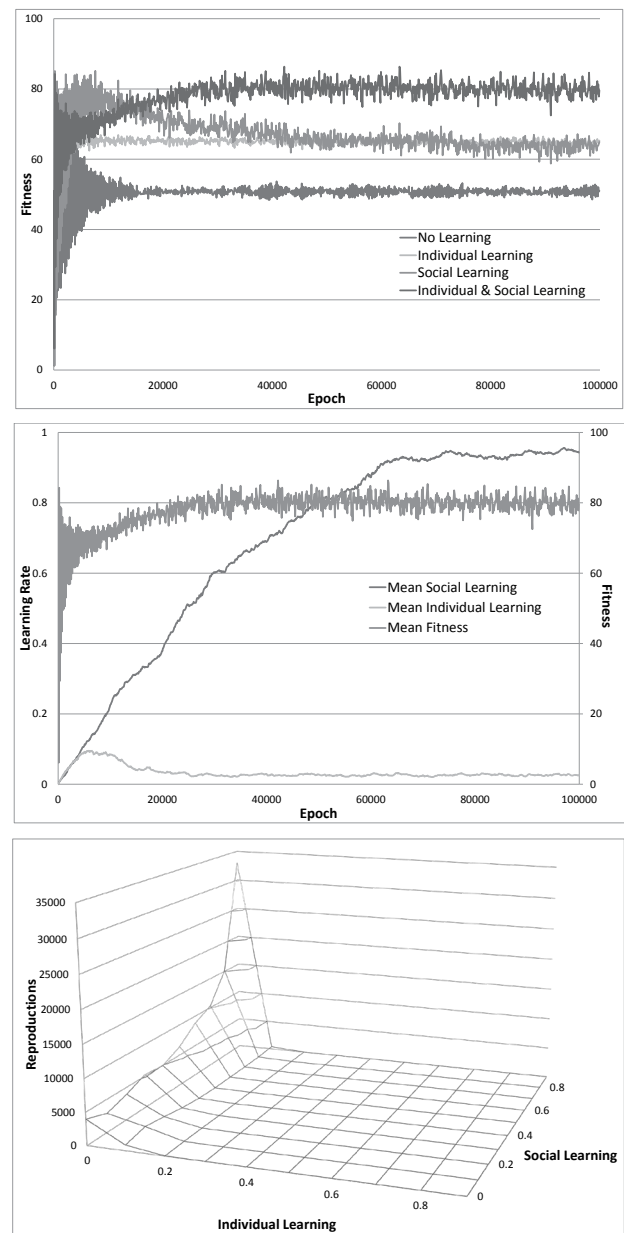


Figure 2: Consistently Variable Environment Tests: a) Mean fitness of each learning strategy, b) Mean fitness of individual and social learning with the evolved learning rates, c) Reproductive fitness of combined learning rates.

ing levels of variability (see figure 3(a)). As the environment becomes more noisy individual learning rates begin to increase, possibly to reintroduce an element of personal innovation to the population, which has become stagnant due to the high levels of conformist learning imposed by large quantities of social learning during times of minimal variability. The reproductive fitness of individuals, as seen in figure 3(c), is also interesting, as reproductively successful

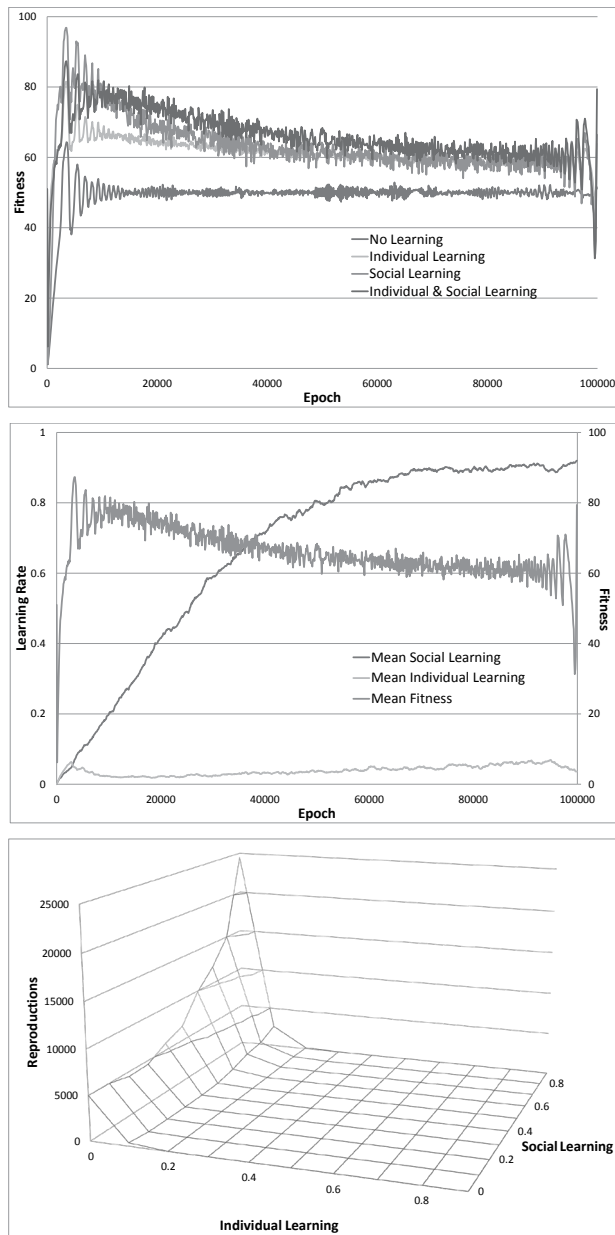


Figure 3: Increasingly Variable Environment Tests: a) Mean fitness of each learning strategy, b) Mean fitness of individual and social learning with the evolved learning rates, c) Reproductive fitness of combined learning rates.

individuals tend to exhibit high levels of social learning and increased levels of individual learning, when compared to the reproductive fitnesses of individuals in consistently variable or static environments. It is also interesting to note the comparisons between maximum learning rates for social and individual learning on increasingly variable environments (see figure 4): despite individual learning being a necessary component of the combined strategy, it is not

exhibited to as high a degree as when found alone; conversely social learning is always exhibited at higher levels when accompanied by individual learning. This again suggests that, while social learning is adopted over individual learning, individual learning is necessary for social learning to be used to greatest effect (Acerbi and Nolfi, 2007; Acerbi et al., 2007). Evidence from all stages of environmental variability seem to tell a similar story, though to different degrees: social learning is widely adopted over individual learning when found together, with all extra-genetic learning strategies performing better than random on all tests. Extra-genetic learning strategies are also exhibited at higher levels in noisy environments than in static environments. The evidence presented does suggest that increasing variability is sufficient to cause the adoption of versatile survival strategies such as learning, with social learning being the learning strategy of choice.

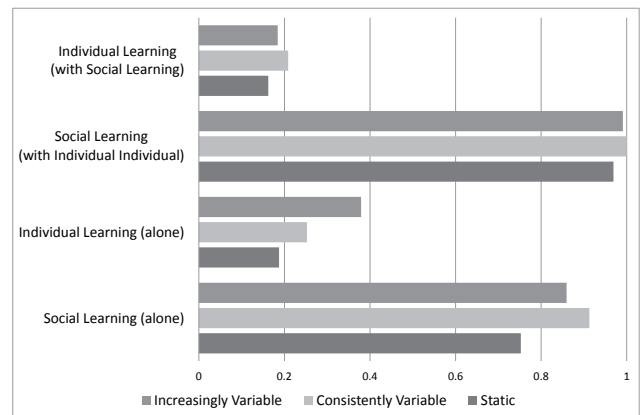


Figure 4: Maximum learning rates exhibited over all environmental test cases for all learning strategies.

Population Collapse in Variable Environments (Consistent and Increasing)

One of the pitfalls of the kind of genetic algorithm used so far is that even when populations exhibit low levels of evolutionary proficiency, they still survive; of course this is not the case in nature. To explore whether or not the learning strategies implemented in this model are really robust we have also implemented a set of tests where populations may become extinct. The first tests follow the test setups above, with populations exhibiting different learning strategies being tested on environments with consistent and increasing variability. Populations falling below $N \times 0.1$ individuals are considered as being collapsed.

Consistently variable environments were produced with four levels of variability;

1. No variability (static): $f = 0$
2. Low variability: $f \in N(0.018, \frac{0.018^2}{2})$

3. Medium variability: $f \in N(0.18, \frac{0.18^2}{2})$

4. High variability: $f \in N(1.8, \frac{1.8^2}{2})$

Learning Strategy	Static	Low	Medium	High
Genetic	100%	100%	100%	0%
Individual	100%	100%	100%	50%
Social	100%	100%	90%	0%
Individual & Social	100%	100%	100%	0%

Table 1: Consistently Variable Environments: % of populations surviving.

The percentages of populations surviving until the end of evaluation are reported in table 1. As may be expected, populations are unable to survive highly variable environments as the increased chance of death makes it all but impossible to re-adapt to new environments. However, individual learning does seem to be more robust than all other strategies, achieving a 50% survival rate on high frequency environments. It may be the case that higher rates of individual learning, though risky, are better able to deal with sudden environmental shifts. Social learning on the other hand begins to struggle in environments exhibiting medium amounts of variability. As with our earlier tests it may simply be the case that conformism spreads through the population, increasing the likelihood of population collapse. Combining individual and social learning alleviates the problem to some extent.

Increasingly variable environments were produced at three initial levels of variability: static, low and medium. In these environments variability increase throughout evolution, to a level of high variability.

Learning Strategy	Static	Low	Medium
Genetic	0%	0%	0%
Individual	100%	100%	100%
Social	0%	0%	0%
Individual & Social	0%	0%	0%

Table 2: Increasingly Variable Environments: % of populations surviving.

Unlike in consistently variable environments all learning strategies, excluding individual learning alone, result in populations that are unable to survive in any increasingly variable environment (see table 2). It seems social learning completely undermines individual learning when combined, perhaps owing to over-conformism in times of lower variability stagnating the population's pool of knowledge to the point that the increase in individual learning, usually seen later in increasingly variable environments (see figure 3(b)) is insufficient to redeem the population's fortunes.

As indicated by tables 1 and 2, individual learning is the only learning strategy robust enough deal with increasing

and high levels of environmental variability. However, in early tests the combined strategy of both individual and social learning was seen to be adaptive in all environmental settings. To investigate whether individual learning is necessary for the successful introduction of social learning we implemented a final set of tests. In these, individual learning was allowed to evolve in isolation for 100000 epochs before the introduction of social learning alongside it for a further 100000 epochs. These tests provide a greater challenge for populations as they are required to survive for twice the evaluation period previously tested. However, this increase in evaluation time does reduce the rate at which environmental variability increases during increasing-variability tests.

As table 3 shows, the evolution of individual learning prior to social learning does provide some benefits in increasingly variable environments, but only when beginning from medium levels of variability ($f = N(0.18, \frac{0.18^2}{2})$). It may be that noisier environments provide a greater selection pressure for high levels of innovation, which in turn introduces a larger pool of knowledge for social learning to access; or that the lower rate of increase in variability is significant. Further tests will need to be conducted to analyze the precise learning rates, reproductive fitnesses and death rates exhibited in these “*goldilocks*” conditions.

Variability	Static	Low	Medium	High
Consistent	100%	100%	100%	0%
Increasing	0%	0%	100%	N/A

Table 3: Individual and Social Learning: % of populations surviving when individual learning is allowed to evolve before the introduction of social learning.

Conclusions and Future Work

Reader and Laland (2002) have demonstrated that personal innovations (individual learning) and social learning co-vary across species. The above results go some way to explaining why social learning was adopted most strongly when combined with individual learning. It seems that individual learning is necessary for effective social learning. This may also be a mechanism of avoiding population collapse. Whilst social learning alone can maintain adaptive knowledge in the population, over-reliance on it can just as easily reinforce sub-optimal or incorrect knowledge when the environment is highly stochastic, potentially causing the population to collapse (Whitehead and Richerson, 2009). By maintaining a level of personal innovation alongside social learning, populations can maintain non-conformist local search whilst ensuring that useful innovations are transmitted over generations (Acerbi and Nolfi, 2007). However, in environments of lower variability conformist social learning ensures a high level of individual fitness. Individual learning on

the other hand may impose unnecessary local search which could cause individuals to lose useful adaptations if high levels of individual learning are maintained. The data presented here suggests that when environments are in minimally variable states individual learning plays a smaller role than it does in more variable environments. It is also found to be the case that mortality is greatly increased in environments of high or increasing variability when social learning is exhibited unless individual innovation is allowed to develop in isolation (Acerbi et al., 2007).

Our initial hypothesis (developed in order to test Potts's variability selection hypothesis), that when individual and social learning rates are evolved simultaneously, both increasing and consistently variable environments are sufficient for the adoption of social learning over individual learning, holds true here, though with two main caveats: individual learning is required for successful social learning, and population collapse may only be avoided when individual learning is allowed to pre-evolve in already noisy environments before the introduction of social learning. Both of these caveats require further investigation in steady state genetic algorithms, neural networks (Curran and O'Riordan, 2007) and grounded animat simulations (Borg et al., 2011).

The way noise is implemented also requires further investigation. Sine waves, though used elsewhere to produce environmental variation (Grove, 2011), are not the only pattern of environmental variability found in nature. Further tests could include empirically derived data sets (Grove, 2011) or *red noise* (Whitehead and Richerson, 2009).

References

- Acerbi, A., Marocco, D., and Nolfi, S. (2007). Social facilitation on the development of foraging behaviors in a population of autonomous robots. In Almeida e Costa, F., Rocha, L., Costa, E., Harvey, I., and Coutinho, A., editors, *Advances in Artificial Life*, volume 4648 of *Lecture Notes in Computer Science*, pages 625–634. Springer Berlin / Heidelberg.
- Acerbi, A. and Nolfi, S. (2007). Social learning and cultural evolution in embodied and situated agents. In *Proceedings of the First IEEE Symposium on Artificial Life*, ALIFE'07, pages 333–340. IEEE Press.
- Best, M. L. (1999). How culture can guide evolution: An inquiry into gene/meme enhancement and opposition. *Adaptive Behavior*, 7(3–4):289–306.
- Borg, J. M., Channon, A., and Day, C. (2011). Discovering and maintaining behaviours inaccessible to incremental genetic evolution through transcription errors and cultural transmission. In *Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems*, ECAL'11, pages 102–109. MIT Press.
- Boyd, R. and Richerson, P. J. (1983). The cultural transmission of acquired variation: Effects on genetic fitness. *Journal of Theoretical Biology*, 100(4):567–596.
- Boyd, R. and Richerson, P. J. (1995). Why does culture increase human adaptability. *Ethology and Sociobiology*, 16:125–143.
- Curran, D. and O'Riordan, C. (2007). The effects of cultural learning in populations of neural networks. *Artificial Life*, 13(1):45–67.
- Grove, M. (2011). Speciation, diversity, and mode 1 technologies: The impact of variability selection. *Journal of Human Evolution*, 61(3):306–319.
- Hinton, G. E. and Nowlan, S. J. (1987). How learning can guide evolution. *Complex Systems*, 1:495–502.
- Jones, D. and Blackwell, T. (2011). Social learning and evolution in a structured environment. In *Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems*, ECAL'11, pages 380–387. MIT Press.
- Laland, K. N. (2004). Social learning strategies. *Learning & Behavior*, 32(1):4–14.
- Nolfi, S. and Floreano, D. (1999). Learning and evolution. *Autonomous Robots*, 7(1):89–113.
- Potts, R. (1996). Evolution and climate variability. *Science*, 273(5277):922–923.
- Potts, R. (1998a). Environment hypotheses of hominin evolution. *Yearbook of Physical Anthropology*, 41:93–136.
- Potts, R. (1998b). Variability selection in hominid evolution. *Evolutionary Anthropology*, 7(3):81–96.
- Reader, S. and Biro, D. (2010). Experimental identification of social learning in wild animals. *Learning and Behavior*, 38(3):265–283.
- Reader, S. M. and Laland, K. N. (2002). Social intelligence, innovation, and enhanced brain size in primates. *PNAS*, 99(7):4436–4441.
- Tattersall, I. (2009). Human origins: Out of africa. *PNAS*, 106(38):16018–16021.
- Tomasello, M. (1999). The human adaptation for culture. *Annual Review of Anthropology*, 28:509–529.
- Whitehead, H. and Richerson, P. J. (2009). The evolution of conformist social learning can cause population collapse in realistically variable environments. *Evolution and Human Behavior*, 30(4):261–273.
- Whiten, A. and van Schaik, C. (2007). The evolution of animal cultures and social intelligence. *Phil. Trans. R. Soc. B*, 362(1480):603–620.

Evolutionary Design and Experimental Validation of a Flexible Caudal Fin for Robotic Fish

Anthony J. Clark, Jared M. Moore, Jianxun Wang, Xiaobo Tan and Philip K. McKinley

BEACON Center for the Study of Evolution in Action
Michigan State University
East Lansing, Michigan, USA 48823
ajc@msu.edu

Abstract

Designing a robotic fish is a challenging endeavor due to the non-linear dynamics of underwater environments. In this paper, we present an evolutionary computation approach for designing the caudal fin of a carangiform robotic fish. Evolutionary experiments are performed in a simulated environment utilizing a mathematical model to approximate the hydrodynamic motion of a flexible caudal fin. With this model, time-consuming computational fluid dynamic simulations can be avoided while maintaining a physically realistic simulation. Two approaches are employed to maximize a robotic fish's average velocity. First, a hill-climbing algorithm is applied to find the optimal stiffness for a fixed shape caudal fin. Next, both fin stiffness and shape are simultaneously optimized with a genetic algorithm. Additionally, simulated caudal fins are compared to physically validated fins, which were fabricated with the aid of a 3D printer and tested on a robotic fish prototype. Results show a correlation between evolved results, model predicted behavior, and physical robot performance with some disparity due to the difficulty in accurately approximating real world performance in a simulation environment. Despite the disparity, evolutionary design is shown to be a viable process.

Introduction

Inspired by natural systems, roboticists have modeled robotic fish with the expectation that they will be as efficient and capable as biological fish. Yet, as is the case with many biomimetic systems, robots are not as proficient as their biological counterparts; the materials and electromechanics that make up a robotic fish simply are not as effective as organic tissue. However, robotic fish do have several advantages over other underwater vehicles types such as propeller-driven robots. First, fewer moving components are necessary, which provides additional space for sensors and reduces power requirements. Additionally, a true-to-life appearance may be less intrusive to the inhabitants of a natural ecosystem. Given these characteristics, robotic fish find applications in scenarios ranging from ecological monitoring to biological studies.

The primary obstacle to developing robotic fish can be attributed to domain uncertainty. Aquatic environments are

highly non-linear, which makes the design process a challenging endeavor. For this reason, mathematical models of the hydrodynamic interactions encountered in such environments can improve the design process by providing a means to test design theories. Even with a perfect mathematical model, however, the design process remains a challenge due to the large number of parameters involved in producing realistic motion. Every combination of different materials and electromechanical constraints will produce different performance and requires detailed knowledge of material properties. For example, to fabricate a flexible caudal fin it is necessary to know the modulus of elasticity of the target material. In view of this complexity, it is desirable to create an automated design process that can handle the high-dimensionality of the problem.

Evolutionary computation techniques (genetic algorithms, neuroevolution, genetic programming, and so on) are well suited to such high-dimensional problems. By broadly sampling the solution space, evolutionary algorithms are able to test for and blend the beneficial aspects of unique solutions in order to create efficient mixtures. By integrating a mathematical model into the evaluation phase of an evolutionary algorithm, the idiosyncrasies of an aquatic environment can be exploited to produce effective, even novel, solutions. From such solutions, roboticists can then gain insight into what constitutes a *good* robotic fish design.

In this paper, we propose an evolution-based methodology for the design of a robotic fish caudal fin. Evolutionary optimization occurs in a rigid-body dynamics engine that incorporates a mathematical model of the hydrodynamics associated with a caudal fin. Simulated solutions are first compared to mathematical predictions; a hill-climber algorithm optimizes the stiffness of a fixed shape fin, and the fitness landscape is compared to one derived directly from the model. Next, results are validated by physically realizing a set of fins and testing them on a robotic fish prototype. Fins are fabricated and tested with the aid of a 3D printer and an aquatic test environment. Finally, an evolutionary algorithm is used to optimize the physical characteristics of the caudal fin. Specifically, the stiffness and dimensions

of a rectangular caudal fin are simultaneously evolved for a given control pattern. The chief contribution of this work is an evolutionary design method based on recently developed dynamic models that can be adapted into a general robotics engineering process.

Background and Related Work

Robotic fish have practical applications in the study of natural fish morphology and behavior as well as in ecological monitoring. They can provide researchers with controllable imitations to assess the behavior of real fish (Faria et al., 2010), or they can be used in the study of natural evolution and other biological hypotheses (Long et al., 2006, 2011). Recent work, in which robotic fish interact with golden shiners, has shown that a tethered robot with a movable caudal fin can elicit schooling behavior from a natural fish in a water-flow tank (Marras and Porfiri, 2012). When the tail structure remained stationary, however, the live fish did not respond with a schooling behavior, supporting the hypothesis that a biomimetic robot can aid in fish behavioral research. As demonstrated by that work, fish can interact with a realistic robot as if it were a natural fish. With increasingly sophisticated designs, new insight into fish behavior can be gained that would be impossible by simply observing biological fish in the wild or a static lab environment. Aside from biological studies, robotic fish have been proposed as a platform to monitor environmental conditions (Tan et al., 2006), including activities such as oil spill monitoring in the Gulf of Mexico and surveying oxygen content of inland lakes. As robots more closely resemble natural fish, it may be possible to deploy them as mobile sensor platforms that do not disturb local ecosystems.

Research into fin design and fabrication has focused primarily on modeling fin structures found in nature. Each type of swimming locomotion (for example, anguilliform and carangiform) requires a mathematical model to accurately describe the governing dynamics. A ribbon-like fin on a robot with a series of actuators connected by a malleable material has been shown to be capable of replicating the thrust of real fins (Epstein et al., 2006). Further research (Hu et al., 2009; Mason and Burdick, 2000; Chen et al., 2010; Tan et al., 2010) has yielded insight into carangiform fish locomotion, in which forward propulsion is predominantly generated by the caudal fin. Recently, a mathematical model has been proposed to encompass the different aspects of locomotion that apply to a flexible carangiform caudal fin (Wang et al., 2011, 2012).

Morphological evolution has been the focus of an abundance of studies beginning with Sims's evolution of virtual creatures (Sims, 1994). A major hurdle to any simulation-developed solution is how well it transfers into a physical robot. A so-called "reality-gap" arises when solutions that appear to work well in a simulated environment face issues in a physical environment that were either unforeseen or in-

correctly modeled (Brooks, 1992; Jakobi, 1998; Koos et al., 2010). Approaches to address this problem include evolving the simulator in conjunction with a robot (Bongard and Lipson, 2004) and directly rewarding solutions for performing similarly in reality and simulation (Koos et al., 2010). In the latter approach, only solutions that have a high transferability (a low disparity between simulation and reality) are deemed highly fit. Further narrowing of the gap is possible by developing accurate models for environmental conditions. In (Gomez and Miikkulainen, 2003), for instance, the authors demonstrated that a detailed simulator can be combined with an evolutionary algorithm to produce controllers for finless rockets, which operate in highly non-linear environments. Recently, the reality gap has expanded to include material properties and their response to specific environmental conditions. Since modeling such interactions at the molecular level is presently intractable, our approach is to integrate evolutionary computation with rigorous mathematical modeling of material properties. Whereas evolutionary computation guides the overall process, engineering is needed to model how constituent materials behave when forces are applied to them, enabling accurate evaluation of the robot in simulation.

Methodology

To create such an environment, we built our simulator on top of a mathematical model and an open source rigid-body dynamics engine, the Open Dynamics Engine (ODE) (Smith, 2012). Additionally, to ensure that results are meaningful, we validated our simulator against fins that were physically tested on a robotic fish prototype.

Mathematical Model

Using rigid-body dynamics, natural caudal fin motion can be approximated by dividing the fin into multiple discrete segments connected by a spring and damping system (Wang et al., 2012). Still, the fluidic motion of a fin during locomotion can be hard to model in simulation and equally as hard to replicate on a physical robot. However, with the advent of 3D printers, we can rapidly test a variety of different materials and discover which are most capable of approximating that motion. Lighthill's Elongated Body Theory of Locomotion (Lighthill, 1971) was proposed to describe the movement patterns of a real fish as if the entire body were flexible. In Lighthill's approach, the movement at any point on a body can be approximated using equations that result in the thrust and movement of that point.

All of the fins in this study were rectangular; we are considering other shapes in our on going investigations. The mathematical model we use to compute the forces produced by rectangular fins is based on Lighthill's theory. In this model, a caudal fin is divided into equal-sized segments and the hydrodynamic forces are evaluated independently for each segment along with an additional force acting at the

tip (Wang et al., 2012). The fin segments in the mathematical model are assumed to be connected through a series of spring and dampers that result in a flexible fin structure, as shown in Figure 1.

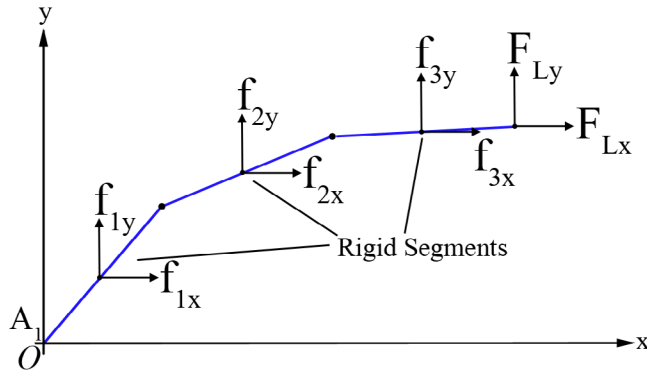


Figure 1: Visual representation of the mathematical model describing the forces acting on the segments of a passive flexible caudal fin.

In the figure, three segments are shown along with the forces that apply to each individual segment. According to the mathematical model, each fin segment generates two component forces, a resistive component and a propulsive component. Each segment experiences hydrodynamic forces described by Equation 1:

$$\vec{f}(\tau) = \begin{pmatrix} f_X(\tau) \\ f_Y(\tau) \end{pmatrix} = -m \frac{d}{dt} (v_{\perp} \hat{n}), \quad (1)$$

where m denotes the mass per unit length, τ is the location on the fin where the force acts, and \hat{n} and v_{\perp} , respectively, are the unit direction and velocity perpendicular to the fin. The tip of the final segment experiences an additional force described by Equation 2:

$$\vec{F}_L = \begin{pmatrix} F_{LX} \\ F_{LY} \end{pmatrix} = \left[-\frac{1}{2} m v_{\perp}^2 \hat{n} + m v_{\perp} v_{\parallel} \hat{n} \right]_{\tau=L}, \quad (2)$$

where $\tau=L$ represents the posterior end of the fin, and \hat{n} and v_{\parallel} , respectively, are the unit direction and velocity parallel to the fin. These hydrodynamic forces can be calculated given the X and Y of each fin segment over time.

At the base of the fin, which is attached to the body, a motor drives the rhythmic motion in a sinusoidal pattern. The parameters for this sinusoidal motion includes the amplitude, frequency, and bias. Along with a material's dimensions, the Young's modulus of elasticity determines flexibility, which is captured in the parameters for the springs and dampers. This relationship provides a means of transferring simulated designs into real materials using known and inferred properties of materials.

Simulation Environment

In view of the unique challenges associated with modeling the fluid dynamics of an aquatic environment, ODE

was used in conjunction with the above mathematical model to approximate the hydrodynamic forces acting on a caudal fin. This method avoids costly computational fluid dynamics calculations. The reduction in computation time is particularly advantageous for evolutionary experiments in which thousands of solutions must be simulated. Consistent with surface-swimming robots, the mathematical model constrains motion to a two-dimensional plane and assumes neutral buoyancy.

The simulated robotic fish is modeled after a physical robotic fish prototype, which was originally constructed to test the performance of different fin dimensions and material stiffnesses. A representation of the virtual model can be seen in Figure 2, showing the main body and a three-segment caudal fin. Fin flexibility was approximated with passive hinges between fin segments governed by predefined spring and damper constraints. This spring system allows the fin to flex at different rates depending on spring and damping coefficients. Rotational movement of the fin is achieved through an actuated hinge connecting the body and first fin segment. The body-fin joint oscillates at 0.9Hz in a 30 degree symmetrical range of motion.

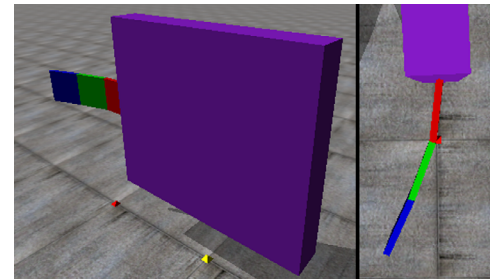


Figure 2: Depiction of the virtual fish model with a three-segment rigid-body caudal fin.

Physical Validation

To validate the proposed method, test fins were fabricated using an Objet Connex350 multi-material 3D printer. Fins were printed with a combination of different physical materials to yield flexibilities that resemble the motion observed in simulation. As demonstrated in (Richter and Lipson, 2011), a 3D printer can considerably improve the efficiency of an experimental design process. Several iterations of printed parts can be fabricated in a matter of hours. The printed fins were attached to a robotic fish prototype and evaluated in an aquatic test environment. An image of the physical robot with attached fin is shown in Figure 3.

Time trials were used to determine the average velocity achieved by each fin, while visual observations helped determine the flexibility of fins during movement. In these physical trials, the height, length, and thickness of each fin were fixed at 2.5, 8.0, and 0.1 cm, respectively. The Young's modulus of elasticity was provided by the manufacturer data sheets. For each of the printed fins, the robot was placed in

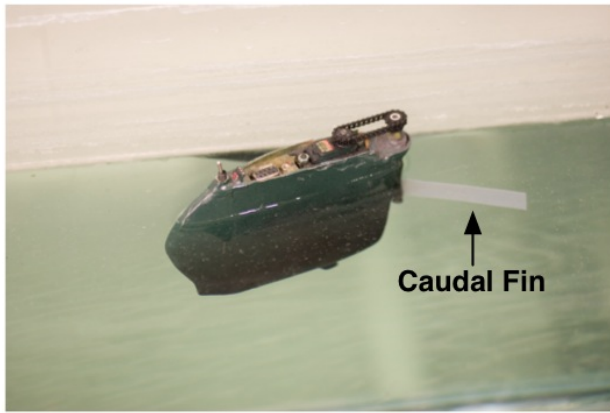


Figure 3: The robotic fish prototype. Movement of the 3D-printed rectangular caudal fin is accomplished using a servo motor with a set range of motion and period of oscillation.

a test tank and allowed to reach a stable swimming speed before the average velocity was computed. The stiffness of each fin can be calculated with Equation 3:

$$K_s = \frac{Edh^3}{12l}, \quad (3)$$

where K_s represents a material's torsion spring constant, d and l denote the height and length of the fin, respectively, E represents Young's modulus of elasticity for the material itself, and h is the thickness of a fin. These values can be directly used in simulation during optimization trials and provide a means of effectively comparing simulation and physical results.

Experiments and Results

The methodology proposed in this paper can be divided into three separate parts: mathematical model validation, physical validation, and evolutionary optimization. We first compared our simulation results with data derived directly from the mathematical model. Next, we performed a similar comparison between simulation and data gathered from physical experiments. Once our simulation environment was validated, we applied evolutionary computation techniques to a flexible fin design process.

Mathematical Model and Simulation

Prior to physical validation and evolutionary experiments, it was important to ensure that our simulation environment matched the mathematical model. Any disparity between simulation and model could signify an error that would make evolutionary results meaningless. With this in mind, two algorithms were employed to optimize the stiffness of the simulated caudal fin. In both experiments, only the Young's modulus was allowed to change.

The first algorithm was a basic hill-climber. For this experiment, 100 independent runs were conducted. Every run

was initialized with a different seed and a Young's modulus value chosen uniformly at random from the range [0, 5 GPa]. Every Young's modulus value was evaluated by translating it, with Equation 3, to the spring coefficients that govern caudal fin flexibility. Once the simulated robotic fish was configured, it was allowed to swim for 10 seconds. The fitness of each Young's modulus was computed as the average velocity achieved over this evaluation period. Each hill-climber run began with the evaluation of the randomly-chosen initial Young's modulus value. Subsequent values were generated by displacing the current value by a random number chosen uniformly from a Gaussian distribution with a mean of 0 and a variance of 0.1. The resulting Young's modulus was then evaluated, and the better performing (higher average velocity) value was kept and used to generate the next test case. In each run, this process was repeated until 100 candidate values had been evaluated. Every hill-climber instance converged to an optimum Young's modulus of roughly 1.9 GPa, and given enough time it is suspected that all final values would converge to a single optimal value.

The second algorithm deployed was a conventional genetic algorithm. The primary use of this experiment was to confirm that the simulation environment could be used effectively with an evolutionary algorithm. This experiment comprised 30 independent runs. Each run was seeded with a different value and a population of 125 randomly generated individuals. Every individual was evaluated in a process identical to that used in the hill-climber experiment. The populations were evolved for 100 generations with mutation as the only evolutionary operator. After population initialization, subsequent generations were created by using a three-individual tournament selection process and a Gaussian mutation operator (identical to the hill-climber displacement operator). Additionally, to ensure that the highest fitness individuals were not lost, the most fit 10% of the population was considered elite and copied to the next generation without modification.

Results from the evolutionary experiment closely resembled those of the hill-climber, with the most fit individuals, in every run, having a Young's modulus near 1.9 GPa. Data generated from the mathematical model can be seen in Figure 4, and results from the two simulation experiments are shown Figure 5. The experimental results show that both the hill climber and evolutionary approaches yield near identical solutions (i.e. a Young's modulus of 1.9 GPa). This is an expected result, as both experiments rely on the same simulation environment.

Comparing Figures 4 and 5, a disparity between model and simulation results is apparent. Specifically, the model predicts a maximum velocity of roughly 5.1 cm/s at a Young's modulus near 0.9 GPa, while simulation results achieve a maximum average velocity closer to 1.4 cm/s at a Young's modulus near 1.9 GPa. Despite the differences,

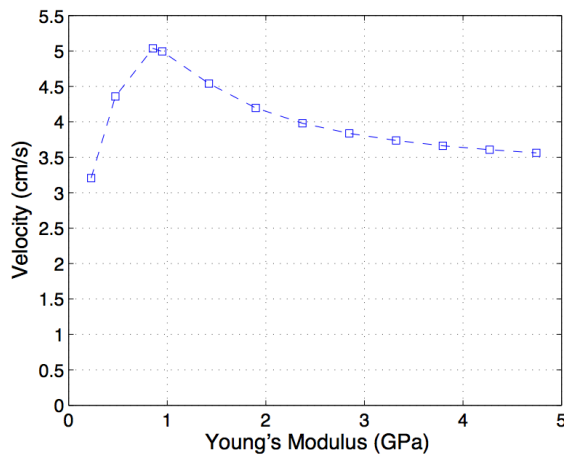


Figure 4: Predicted velocities for different Young's Modulus values from the mathematical model calculations. Note that this assumes that the body is anchored.

both figures show the same trend, in which intermediate values of the Young's modulus produce the fastest robotic fish. Additionally, the disparity between figures can be explained by closer examination of the model and simulator. The most marked differences are that the mathematical model assumes the robotic fish body does not affect caudal fin motion, and the caudal fin segments are without mass. Neither of these assumptions is carried over into the simulation environment, and both of these factors would cause simulated robotic fish to appear *slower* than model data would predict. In the next section, physical results will be examined to determine whether the simulation results are physically meaningful.

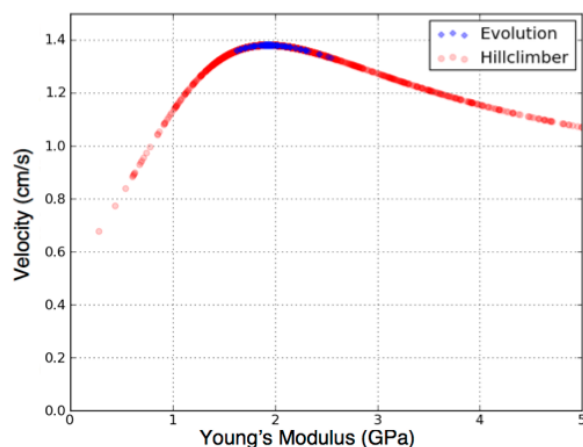


Figure 5: Results of the hill climber and evolutionary runs for determining the optimum stiffness of a fixed dimension fin. Both methods converged on a common stiffness yielding the highest average velocity. Darker shades indicate clustered results from different trials.

Physical Validation

To validate observations taken from simulation, we fabricated caudal fins with a 3D printer and tested them on a robotic fish prototype in an aquatic environment. Six unique fins were printed, each with a different Young's modulus. The materials ranged from extremely flexible (TangoBlack-Plus) to nearly inflexible (VeroWhite). Each printed fin was attached to the robot and tested in the aquatic environment; the average velocity was measured over 5 separate trials. The results of this experiment are plotted in Figure 6. Consistent with the predicted performance, the plot shows that an intermediate flexibility produces the highest average velocity. However, direct comparisons between simulation and reality are not possible due to current limitations of the 3D printed materials. Specifically, the materials do not have an exact Young's modulus value, but rather the manufacturer provides a range of possible values for each material (materials properties are not guaranteed to remain constant between print jobs). For example, VeroWhite has a modulus in the range of 2-3 GPa, while the other materials have lower-value ranges.

In view of the fact that the mathematical model, simulation, and physical data are all for fins of identical shape, some comparisons can yet be made. First, the velocity values of the physical robotic fish are closer to mathematical model predictions than they are to simulation results. The data collected from these experiments will be vital in improving the model and simulation environment. In addition, the optimal Young's modulus for all results is in the range of 1-2 GPa. The reason for the disparity in the model predictions was discussed in the previous section, however it is also apparent that simulation results do not perfectly match reality. The maximum velocity of 3.7 cm/s in the physical experiments is nearly twice the maximum simulation velocity. As with the model, certain approximations were made in the simulation environment. For instance, distributed forces were treated as single point forces, and the flexible fin was split into just three segments. By decreasing the size of each segment and increasing the number of segments, the motion and discretization of forces will be more realistic and likely increase the accuracy of the simulation.

As a secondary measure of performance between the simulation and physical experiments, we observed the flexibility of fins as they oscillated. Figure 7 presents a side by side comparison between a simulated flexible caudal fin and the 3D printed version on the robot. Both series of images display the flexibility of a fin as it oscillates. This visual observation helps to reinforce the viability of simulating flexible caudal fins.

Evolution of Fin Morphology

Upon completion of comparisons between mathematical model and simulation results, optimization was expanded into a full evolutionary computation run in which the

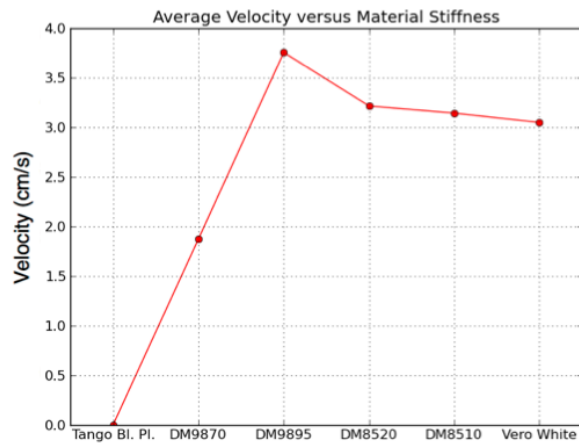


Figure 6: Observed average velocity for different materials used in printed fins. Stiffness increases from left to right in the plot.

Young's modulus and dimensions of a rectangular caudal fin were simultaneously evolved. Fin shape was allowed to evolve under the constraint that the overall area of the length-height face and the thickness of the fin remain fixed. This created a state in which the height of the fin was dependent upon the length of the fin. As such, the two parameters to evolve were the Young's modulus and length of a fin. Practical considerations on the overall dimensions of the fin were also taken into account as a maximum length of 14 cm (length of the robotic fish body) and a minimum length of 4 cm (half the length of previous experiments) were imposed upon evolution. Values outside of this range could suffer from transferability issues given electromechanical constraints such as the maximum torque exerted by a servo. Again, an individual run consisted of 125 individuals evolving for 100 generations. Similar to the previous evolutionary experiments, tournament selection, of size 3, and elitism were used to select the parents for the next generation. Unlike earlier experiments, however, single point crossover was added so that individuals could be generated as a combination of two selected parents. In total, 30 replicate runs were conducted to find the relationship between fin stiffness, fin shape, and average velocity.

From the evolutionary runs, a set of optimum values was found for both the Young's modulus and dimensions of the fin. The Young's modulus found in the trial was 7.55 GPa, and the caudal fin length and height were 14 and 1.43 cm respectively. Hence, the fittest solutions reached the maximum fin length allowed at a cost of fin width. This result was expected, as a longer fin will be able to generate larger propulsive forces, while width has a lesser effect on this force. This characteristic can be seen by close examination of Equation 2, where the length of a fin is a linear factor,

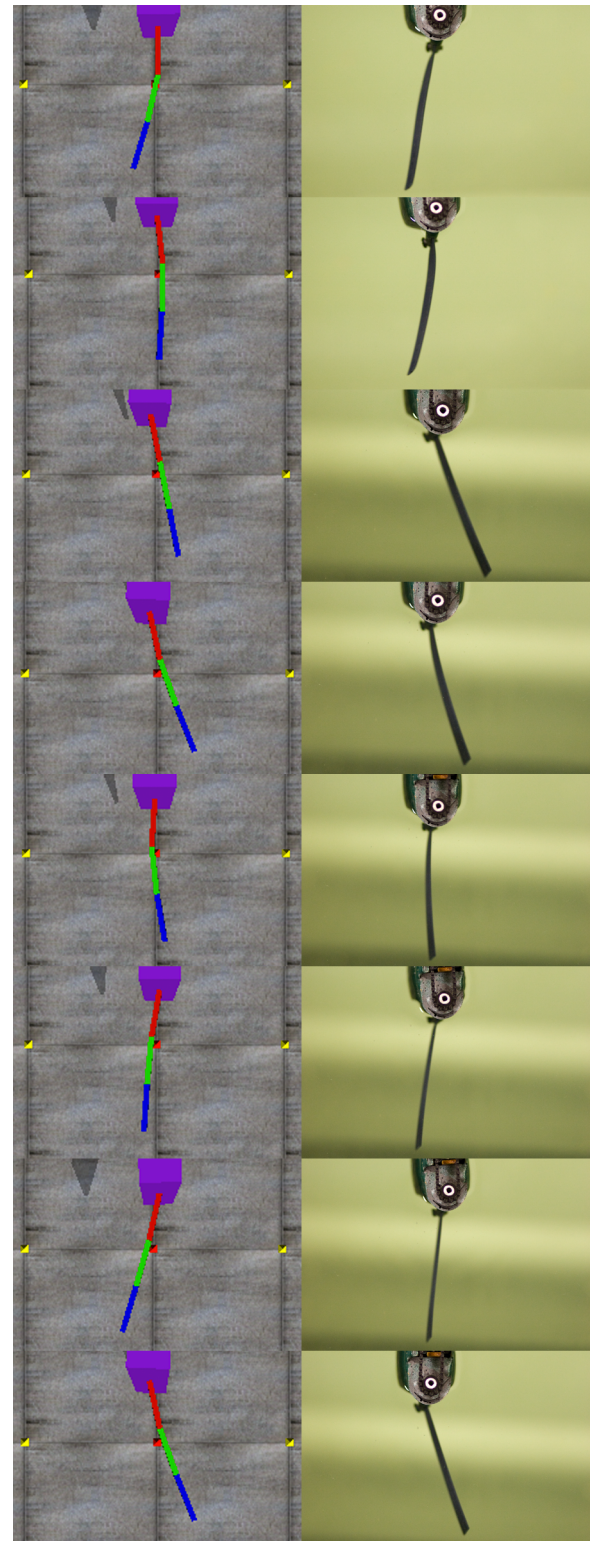


Figure 7: Visual performance of the evolved flexible fin in simulation (left) versus a fabricated flexible fin tested on the prototype robot (right).

and longer fins will have a higher angular velocity near the posterior of the fin.

While the Young's modulus found in the trial is larger than that found in prior experiments, the resulting material stiffness is similar: 1.35×10^{-3} N m for the original experiments, and 1.73×10^{-3} N m for the full evolutionary experiments. This result suggests that a single stiffness value may be adequate for any rectangular caudal fin dimensions. The reason these stiffness values are similar is that as length increased, the Young's modulus also increased to maintain a fairly constant value. Figure 8 presents the three dimensional fitness landscape found in the evolutionary run. As shown, a peak is located at a modulus of elasticity of 7.55 GPa and a length of 14 cm. This combination yielded an average velocity of 2.2 cm/s. This landscape would suggest that for each set of dimensions there is a specific Young's modulus that correlates to the overall best performance for a fin.

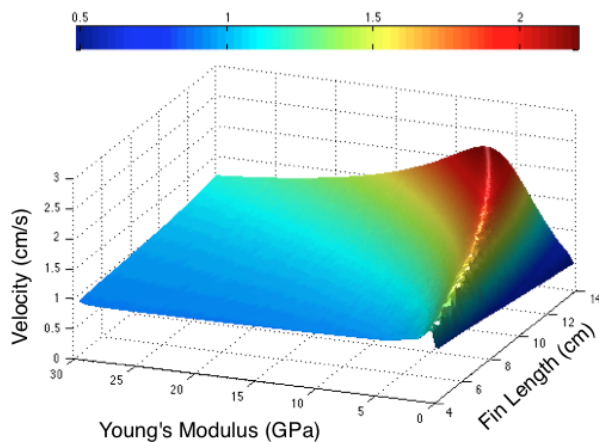


Figure 8: Visualization of the fitness landscape for different shape and stiffness fins. Note that height is dependent upon length in determining shape, therefore, height has been omitted from the data. As the length of the fin increases, the Young's Modulus increases as well to maintain similar stiffness fins for different lengths.

The complex dynamics of an underwater environment make designing efficient robotic fish a challenging engineering endeavor. Considering the difficulty, it is desirable to create an automated design process by which robotic fish can be optimized for a specific task. Making use of the hydrodynamic model for a robotic fish caudal fin, we have shown that an *in silico* process can be used to optimize the Young's modulus of a flexible fin. In simulation, we observed that the optimum Young's modulus is dependent on both the caudal fin motion and dimensions. Specifically, for any combination of fin frequency, amplitude, height, width and length there will be a unique Young's modulus optimum. However,

when the Young's modulus was simultaneously evolved with fin shape, we found that the overall resulting fin stiffness exhibited comparable characteristics. Generally, higher values of length and Young's modulus produced faster swimmers.

Conclusion

In this paper, we demonstrated an evolutionary design method for robotic fish caudal fins. We first developed a simulation environment in which unique fin configurations could be tested. The simulation environment was created by combining a rigid-body dynamics engine with a mathematical model of a flexible caudal fin's hydrodynamics. To test the simulation environment, we first implemented a hill-climber algorithm. Given a fixed fin shape and control pattern, the hill-climber algorithm mapped-out the fitness landscape for fin stiffness vs. velocity. These results were compared to data generated directly from the model, which confirmed that the simulation and the mathematical model have comparable dynamics, although the absolute values differ.

Hill-climber results were further validated through comparisons with physical experiments. With the aid of a 3D printer, an aquatic test environment, and a robotic fish prototype, we conducted a series of velocity tests for several 3D-printed fins. All fins were identical in shape, but had stiffness values (i.e. Young's modulus) ranging from very low to nearly inflexible. Plots of stiffness vs. velocity for the mathematical model, simulation, and physical experiments all showed a similar trend in which average velocity was maximal for intermediate caudal fin flexibility. This result demonstrates that it is possible for a simulation environment to capture key aspects of the dynamics of flexible materials.

To simultaneously optimize several fin parameters, we progressed from the hill-climber experiments to an evolutionary algorithm. A conventional genetic algorithm was used to evolve both the Young's modulus and shape of a fin. From this series of experiments, we found that the most fit fins generally evolved to be as long as possible while maintaining a fairly constant stiffness value. This result is consistent with the fact that longer fins generally produce larger propulsive forces. Additionally, our results showed that for each fin shape and control pattern there is an associated optimal Young's modulus.

The simulated and physical results discussed in this paper demonstrate the effectiveness of an evolutionary based approach given the high dimensionality of the solution space. To continue this research, our future work will focus on improving the design process. First, basic assumptions central to the hydrodynamic model will be removed. For instance, the body will no longer be considered anchored and the fins no longer without mass. Our rigid-body simulator will also be improved by converting our single-point forces to more accurate distributed forces. These improvements alone are likely to increase the accuracy of the simulation and in turn facilitate the transfer of simulated solutions to

reality. Next, we will gradually relax the constraints placed on evolution. In biological fish, caudal fins predominantly increase in height towards the posterior, and accordingly evolution should be allowed to evolve non-rectangular fins. Additionally, due to fin motion being a key component of optimization, it is likely that evolution will be able to find more appropriate control patterns. Ultimately, the goal is to simultaneously evolve as many aspects of the robotic fish as possible in a process that can be generalized to any non-linear robotic environment.

Acknowledgements

The authors gratefully acknowledge the contributions and feedback on the work provided by Professor Janette Boughman, members of the Software Engineering and Network Systems Laboratory, the Smart Microsystems Laboratory, the Digital Evolution Laboratory, and the BEACON Center at Michigan State University. This work was supported in part by National Science Foundation grants CNS-1059373, CNS-0915855, DBI-0939454, CCF-0820220, IIS-0916720, ECCS-1050236, ECCS-1029683 and CNS-0751155; and by U.S. Army Grant W911NF-08-1-0495.

References

- Bongard, J. C. and Lipson, H. (2004). Once more unto the breach: Co-evolving a robot and its simulator. In *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*, pages 57–62, Boston, Massachusetts, USA.
- Brooks, R. A. (1992). Artificial life and real robots. In *Proceedings of the First European Conference on Artificial Life*, pages 3–10. MIT Press, Cambridge, MA.
- Chen, Z., Shatara, S., and Tan, X. (2010). Modeling of biomimetic robotic fish propelled by an ionic polymer metal composite caudal fin. *IEEE/ASME Transactions on Mechatronics*, 15(3):448–459.
- Epstein, M., Colgate, J., and MacIver, M. (2006). Generating thrust with a biologically-inspired robotic ribbon fin. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2412–2417, Beijing, China.
- Faria, J., Dyer, J., Clément, R., Couzin, I., Holt, N., Ward, A., Waters, D., and Krause, J. (2010). A novel method for investigating the collective behaviour of fish: introducing ‘Robofish’. *Behavioral Ecology and Sociobiology*, 64:1211–1218.
- Gomez, F. J. and Miikkulainen, R. (2003). Active guidance for a finless rocket using neuroevolution. In *Proceedings of the 2003 Genetic and Evolutionary Computation Conference*, pages 2084–2095, San Francisco, California, USA. Morgan Kaufmann.
- Hu, Q., Hedgepeth, D., Xu, L., and Tan, X. (2009). A framework for modeling steady turning of robotic fish. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pages 2669–2674, Kobe, Japan.
- Jakobi, N. (1998). Running across the reality gap: Octopod locomotion evolved in a minimal simulation. In *Proceedings of the First European Workshop on Evolutionary Robotics*, pages 39–58, Paris, France. Springer-Verlag.
- Koos, S., Mouret, J. B., and Doncieux, S. (2010). Crossing the reality gap in evolutionary robotics by promoting transferable controllers. In *Proceedings of the 2010 ACM Genetic and Evolutionary Computation Conference*, pages 119–126, Portland, Oregon, USA. ACM.
- Lighthill, M. J. (1971). Large-amplitude elongated-body theory of fish locomotion. In *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 179(1055):125–138.
- Long, J. H., Koob, T. J., Irving, K., Combie, K., Engel, V., Livingston, N., Lammert, A., and Schumacher, J. (2006). Biomimetic evolutionary analysis: testing the adaptive value of vertebrate tail stiffness in autonomous swimming robots. *Journal of Experimental Biology*, 209(23):4732.
- Long, J. H., Krenitsky, N. M., Roberts, S. F., Hirokawa, J., de Leeuw, J., and Porter, M. E. (2011). Testing biomimetic structures in bioinspired robots: How vertebrae control the stiffness of the body and the behavior of fish-like swimmers. *Integrative and Comparative Biology*, 51(1):158–75.
- Marras, S. and Porfiri, M. (2012). Fish and robots swimming together: attraction towards the robot demands biomimetic locomotion. *Journal of The Royal Society Interface*.
- Mason, R. and Burdick, J. (2000). Experiments in carangiform robotic fish locomotion. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, volume 1, pages 428–435.
- Richter, C. and Lipson, H. (2011). Untethered hovering flapping flight of a 3d-printed mechanical insect. *Artif. Life*, 17(2):73–86.
- Sims, K. (1994). Evolving virtual creatures. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’94, pages 15–22, New York, NY, USA. ACM.
- Smith, R. (2012). Open Dynamics Engine, <http://www.ode.org/>.
- Tan, X., Carpenter, M., Thon, J., and Alequin-Ramos, F. (2010). Analytical modeling and experimental studies of robotic fish turning. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pages 102–108, Anchorage, Alaska, USA.
- Tan, X., Kim, D., Usher, N., Laboy, D., Jackson, J., Kapetanovic, A., Rapai, J., Sabadus, B., and Zhou, X. (2006). An autonomous robotic fish for mobile sensing. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5424–5429, Beijing, China.
- Wang, J., Alequin-Ramos, F., and Tan, X. (2011). Dynamic modeling of robotic fish and its experimental validation. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 588–594, San Francisco, California, USA.
- Wang, J., McKinley, P. K., and Tan, X. (2012). Dynamic modeling of robotic fish with a flexible caudal fin. In *Proceedings of the 5th Annual Dynamic Systems and Control Conference*, Ft. Lauderdale, Florida, USA. Accepted for publication.

Informational Drives for Sensor Evolution

Sander G. van Dijk and Daniel Polani

Adaptive Systems Research Group, University of Hertfordshire, Hatfield, UK

Abstract

It has been hypothesized that the evolution of sensors is a pivotal driver for the evolution of organisms, and especially, as a crucial part of the perception-action loop, a driver for cognitive development. The questions of why and how this is the case are important: what are the principles that push the evolution of sensorimotor systems? An interesting aspect of this problem is the co-option of sensors for functions other than those originally driving their development (e.g. the auditory sense of bats being employed as a ‘visual’ modality). Even more striking is the phenomenon found in nature of sensors being driven to the limits of precision, while starting from much simpler beginnings. While a large potential for diversification and exaptation is visible in the observed phenotypes, gaining a deeper understanding of why and how this can be achieved is a significant problem. In this present paper, we will introduce a formal and generic information-theoretic model for understanding potential drives of sensor evolution, both in terms of improving sensory ability and in terms of extending and/or shifting sensory function.

Introduction

An organism may be seen as the result of a possibly large set of trade-offs between different evolutionary pressures. For example, a predator may be driven to become bigger and stronger to enable it to overpower larger prey, while at the same time there may be a pressure towards lighter and leaner bodies, such that it can better outrun its meal. For sensors, such a trade-off is shown for example to exist between spatial and temporal visual resolution (Kortmann et al., 2001), and a similar trade-off is hypothesized for an organism’s cognitive abilities (Polani, 2009): larger brains, and larger or more precise sensors to supply such brains with more detailed input, open up a wider range of behavior, but cognitive facilities that are more complex than necessary to support the organism’s behavior waste vital resources. The significance of the level of energy consumption incurred by sensory and information processing systems is exemplified by multiple studies; e.g. the eye of a resting fly accounts for 10% of its energy consumption (Laughlin et al., 1998), which compares to 20% for the human brain (Kandel et al., 2000). Such insights lead to the expectation that

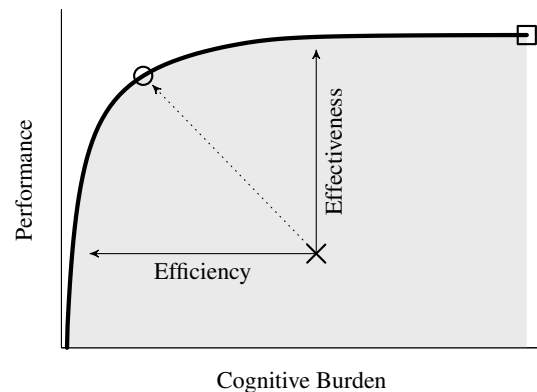


Figure 1: Trade-off between cognitive burden and behavioral performance. The available cognitive power restricts the range of feasible behavioral performance, denoted by the shaded area. The boundary of this area (solid line) traces the *optimal trade-off curve*, i.e. the highest performance achievable without surpassing a given load, or, equivalently, the minimal load needed to achieve a given level of fitness, with the global optimum with the highest performance at the tip (square). A species below this curve will feel evolutionary pressures to be cognitively more efficient, and/or use its cognitive power more effectively (solid arrows), moving it towards a point on the optimal curve (dotted arrow, circle).

organisms are driven to operate on the optimal trade-off between sensory-cognitive burden and behavioral performance (Polani, 2009).

It should be noted that this implicitly assumes an ‘arms race’ of sorts between an agent’s cognitive and behavioral facilities. If an organism does not operate at an optimal trade-off level, we assume there is a drive to increase fitness through more effective utilization of the superfluous cognitive capacity, while another pressure pushes towards degeneration of the sensory and cognitive capabilities to be more efficient and do away with unneeded energy consumption, until these pressures meet in the middle. See also Fig. 1. At this point a so called ‘Pareto-efficient’ optimum is reached,

where a unilateral change in a single component will push the organism away from the optimal trade-off. Moving from one point on the trade-off curve to another would thus need concurrent, well matched evolutionary steps in both sensor and actuation space. Such synchronous, mutually reinforcing steps are highly unlikely, since in a random evolutionary scenario this requires two coordinated mutations. If this reasoning is correct, evolution would be slowed down considerably once a species' sensory-motor system has reached and operates on the optimal trade-off curve.

It is clear from nature however that this is not the case: species evolve continuously, and sometimes at considerable speeds. Species that are optimally adapted to a specific niche still seem able to rapidly specialize for and occupy another niche if the opportunity arises. Even more fascinating is that biological organisms do not seem to evolve simply towards any random locally optimal trade-off, but are instead driven to the *near-global* optima where their sensory capabilities are only limited by the laws of physics. Some striking examples are the retinal receptors of toads that can detect single photons (Baylor et al., 1979), a viper's pit heat sensor that can react to heat differences of 0.003°C (Bullock and Diecke, 1956), and the fact that the inner ear detects forces comparable to the thermal-noise limit (Denk and Webb, 1989).

These considerations lead to the following questions. Firstly, how is it possible that species can evolve quickly from one local optimum to another, while local changes seemingly can only reduce their fitness, without the need of highly unlikely large and coordinated mutations? Secondly, what are possible factors that drive and facilitate sensory evolution towards the ultimate limit of precision?

In the current paper we introduce an information-theoretic framework to help gain insight into these problems. We show 1) how the apparent co-dependence of sensory and actuation systems can be decoupled, 2) how this enables the gradual development of the combined system from one optimum to another, and 3) how this results in strong evolutionary pressure towards maximally advanced sensors.

The use of information-theoretical methods to study life and evolution is becoming increasingly popular. This use is motivated by the view of an agent as an information processing system that is interacting with the environment through a sensory and an actuation channel (Touchette and Lloyd, 2000). Concepts and methods from the field of *Information Theory (IT)* can be applied directly to model and analyze such systems. This kind of modeling can lead to fundamental insights, such as in fundamental limits on control (Touchette and Lloyd, 2004), how embodiment induces information structure in sensory inputs (Pfeifer et al., 2007), exploratory behavior (Ay et al., 2008), and the optimal trade-off between sensory and cognitive burden and performance of an organism (Polani et al., 2006; Tishby and Polani, 2011; van Dijk et al., 2010).

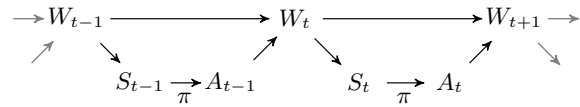


Figure 2: Perception-Action loop as a Causal Bayesian Network. The world state at time t is denoted by the random variable W_t , the resulting sensor state by S_t , and A_t expresses the action taken by the agent. The edges depict the causal interactions between the random variables.

Following these latter works, we correlate the sensory and cognitive burden for an organism with the amount of information that it necessarily needs to take in and process to execute its behavior. As we will show in the remainder of the paper, this implies that the optimal trade-offs will be those where an agent's performance is optimal given its informational burden, or equivalently, where a given level of performance is achieved with the minimal informational requirements.

The major appeal of applying IT to the study of organisms and evolution is that it allows for universal quantitative statements that hold for all systems, both natural and artificial, with only very general assumptions about the properties of the actual realization of, and cognitive mechanisms behind, such systems. This also means that we must stress that, while we believe that this family of methods capture the essence of possible drives for the evolution of sensory-motor systems, we do not wish to claim that the methods used to derive and achieve such limits necessarily accurately reflect the actual mechanisms of natural evolution.

In the following two sections we will introduce the formal frameworks that form the foundation of our approach. Next, we will develop a model of how the evolution of sensors and actuation can be uncoupled to facilitate transition from one locally optimal trade-off to another. We will then adapt this framework to model how evolution could drive sensors towards the upper limits of precision. Finally, we present fundamental information-theoretic properties of sensory systems that facilitate such processes, and argue that these properties constitute major, general, and fundamental drivers of sensor evolution.

Perception-Action Loop

We treat the *Perception-Action loop (PA-loop)* as a *Causal Bayesian Network (CBN)*, shown in Fig. 2, in line with Touchette and Lloyd (2004) and Klyubin et al. (2004). Here, each node is a random variable, which we denote by capital letters (W_t, S_t, A_t), and the edges depict the directional causal interactions between these variables. The set of values that a variable can take is written with corresponding calligraphic capital ($\mathcal{W}, \mathcal{S}, \mathcal{A}$), while small letters are used for concrete instantiations (w_t, s_t, a_t).

In the CBN above, the world state at time t is given by the value $w_t \in \mathcal{W}$ of W_t . This state induces a sensor state $S_t = s_t \in \mathcal{S}$, according to a probabilistic mapping $p(s_t|w_t)$. The agent then selects its action $A_t = a_t \in \mathcal{A}$ based on this sensor state, following a *policy* $\pi(a_t|s_t) = p(a_t|s_t)$. This action, combined with the previous world state, determines the next state of the world according to the *transition probability function* $P_{w_t, a_t}^{w_{t+1}} = p(w_{t+1}|w_t, a_t)$.

This models the agent-world dynamics. We endow these dynamics with a reward structure that determines preferable and less preferable behaviors of the agent. This we do by adopting the standard framework of *Markov Decision Processes (MDP)* (Sutton and Barto, 1998) with a *reward function* $R_{w_t, a_t}^{w_{t+1}}$ that gives the *immediate reward* r_t presented to the agent for the transition of the world state from w_t to w_{t+1} , by performing action a_t . This reward function, combined with a policy, defines a *utility function* over state-action pairs, $U^\pi(w_t, a_t)$, as the expected total reward accumulated by the agent performing an action in a certain state and continuing by following the given policy:

$$\begin{aligned} U^\pi(w_t, a_t) &= E[r_t + r_{t+1} + r_{t+2} + \dots | w_t, a_t, \pi, P, R] \\ &= \sum_{w_{t+1}} P_{w_t, a_t}^{w_{t+1}} \left[R_{w_t, a_t}^{w_{t+1}} + E[U^\pi(W_{t+1}, A_{t+1})] \right], \end{aligned} \quad (1)$$

where

$$\begin{aligned} E[U^\pi(W_{t+1}, A_{t+1})] &= \\ &= \sum_{w_{t+1}} p(s_{t+1}|w_{t+1}) \sum_{a_{t+1}} \pi(a_{t+1}|s_{t+1}) U^\pi(w_{t+1}, a_{t+1}). \end{aligned}$$

In this framework achieving more reward is desirable, and we assume that evolution drives towards policies and sensors that enable higher accumulated rewards. The overall expected total reward, $E[U^\pi(W_t, A_t)]$, can thus be seen as a correlate to an agent's evolutionary fitness. However, this measure alone does not take into account that a policy may require a significant cognitive burden in order to execute. In the following section we extend the framework in order to correct the fitness measure for this.

Information in the PA-Loop

With the concepts of the previous sections, we can develop our framework for the informational treatment of the PA-loop. As mentioned in the introduction, we treat an agent as an information processing system. In other words, an agent takes in a certain amount of information about the world state through its sensors, which it processes to base its action selection on.

The field of Information Theory supplies methods to quantitatively treat such notions about information, and offers strict bounds that such quantities must adhere to. For instance, given a policy, there is a certain amount of information about the world that on average needs to pass through

the agent's sensors and action selection mechanism at each time step to be able to execute that policy. In the model of the PA-loop described above, this amount is quantified by the *mutual information* $I(W_t; A_t)$ between the world-state and action variables. It is argued that this quantity is a major indicator of the cognitive burden imposed on the agent by the policy (Polani et al., 2006), and here we will treat it as such.

In this framework, we can ask for the minimal amount of informational burden required to achieve a fixed level of performance. The answer to this is found by minimizing $I(W_t; A_t)$ over all possible policies $\pi(a_t|w_t)$ (which we will denote a *direct* policy, as opposed to the definition of a policy above that selects an action based on the world state indirectly through a sensor), under the constraint of a fixed performance level $E[U^\pi(W_t, A_t)]$. This can be achieved through an iterative algorithm derived from standard IT methods, as shown by Polani et al. (2006). The minimum amount of information found this way is known as the *Relevant Information (RI)*, as this is the minimal information that is relevant to achieving a certain level of performance. The RI methods can be used to trace out the full optimal trade-off curve, from one extreme where we find the policy that induces the minimal amount of informational burden needed to achieve the absolute maximum level of performance, to the other, where the optimal behavior is found for a 'blind' agent that takes in no information at all; in the current paper we only treat full optimality, and thus always find the first trade-off.

Once we have found such an *RI-optimal* direct policy, we can employ a related IT paradigm, that of the *Information Bottleneck (IB)* (Tishby et al., 1999), to find a *minimally optimal* sensor mapping $p(s_t|w_t)$ for this policy. With this we mean a mapping that is optimal in the sense that it *retains all relevant information* to support a policy $\pi(a_t|s_t)$ that is consistent with the RI-optimal direct policy, and minimal in the sense that it *captures the minimum amount of information* about the world state to be able to reconstruct this information. In other words, the distinctions that the sensor can make between world states must be precise enough to perform the RI-optimal policy, but not more precise than that. Formally, these two requirements mean that we find a sensor that satisfies the constraint $I(S_t; A_t) \stackrel{!}{=} I(W_t; A_t)$, while minimizing $I(W_t; S_t)$.

Uncoupled Sensor-Actuation Evolution

With the formal foundation of our approach in place, we will now develop an evolutionary model in which transitions between different locally optimal trade-offs are made feasible, by uncoupling the evolution of sensors and actuation.

In this model, we start out with an agent whose sensor and action selection mechanism operate on the globally optimal trade-off between informational burden and performance. This trade-off is fully determined by the utility of its actions

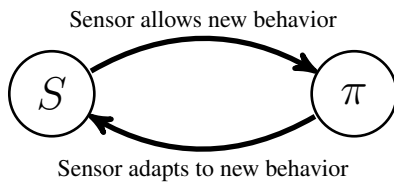


Figure 3: Graphical representation of uncoupled iterative evolution model

and the world dynamics, and can be found using the RI and IB methods discussed in the previous section. As noted before, it seems this point seems to constitute an evolutionary dead-end, even more than any other locally, Pareto-optimal trade-off, since no improvement at all is possible.

Our solution to this problem is based on the idea that, given the currently evolved minimally optimal sensor, there could be other niches available for which this sensor is near-optimal. We will show that this view allows sufficient decoupling of the development of the components, which makes the necessary individual evolutionary steps much more likely.

The basic functioning of this model is visualized in Fig. 3: even when the sensor may be strictly minimal for a policy achieving optimal performance given one reward structure, this sensor may still give enough information to allow successful operation under a *different reward function*, and achievement of a *similar level of fitness* in this new scenario. In that case, evolution can drive the agent's behavior, as expressed by its policy, to become optimal in this new situation, *without* the need of coordinated adaptation of the sensor. Once the transition to this new niche has started, the development of the sensor can instead *follow* that of the action selection mechanism, to again become minimally optimal. Here, we make no explicit assumption of what motivates such a transition between different niches, but possible drives may be toughening competition in the original niche, or perhaps simply evolutionary drift when the fitness achievable in both niches is similar enough.

To clarify this idea, we apply this model to an example from nature of the transformation of a sensor. Tachinid flies possess a balloon-like sensor to detect movement of the head, which in the parasitoid *Therobia leonidei* has been evolved into an auditory sensor, which now is used in locating the bush-crickets that serve as its host (Lakes-Harlan and Heller, 1992). This transformation can be explained in our model by noting that the original sensor, even if it would be fully optimized and minimal for its original use, may capture additional information that is relevant to the organism. In this case, the cognitive and actuation system of the organism can evolve to utilize this information, i.e. to better locate hosts, which constitutes the first step of the cycle above. Once this adaptation is set in motion, the evolution of the sensor can

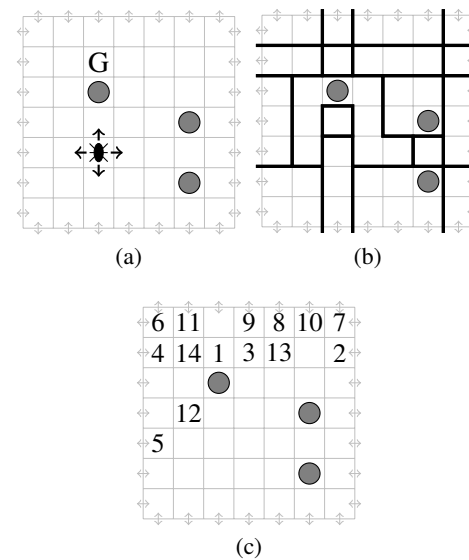


Figure 4: (a) Example 7×7 toroidal grid-world used to demonstrate our model. The world-state consists of the agent's location. The agent receives a penalty of -1 for each step taken, unless it enters the goal state marked G , where reward is 0. The agent has access to 4 actions: move one cell north, east, south or west. Three randomly chosen cells, marked by gray disks, incur a reward of -5 when entered. (b) Location distinctions as given by minimally optimal sensor for task shown in (a). (c) Example of sequence of goals of first 12 tasks in expanding repertoire scenario.

be driven towards higher auditory precision to better support the new strategy, which forms the second step of the cycle. These processes can then repeat until a new local optimum is reached, where the now auditory sensor is minimally optimal for its new function. Note that at no point of this process a coordinated adaptation of the combined sensory-actuation system is needed.

In this paper, we use a simple toroidal grid-world navigation task example, as depicted in Fig. 4, to show how this model works. The notion of different possible niches central to our model, formulated as different reward structures, is in such scenarios represented by a set of tasks, each with its according reward function. Here, each task is described by a goal state g that the agent needs to move into in as few steps as possible, formalized by a reward function that penalizes each step with a reward of -1, unless the agent enters the goal state, where the reward is 0. To prevent trivial solutions due to the high symmetry of the world, and to make lack of information about the world state more costly, several states are marked as 'danger' states that incur a cost of 5 upon entering. A sensor in this world maps, or clusters, world states to a smaller set of sensor states, determining the precision in which the agent can observe its location. Figure 4b shows

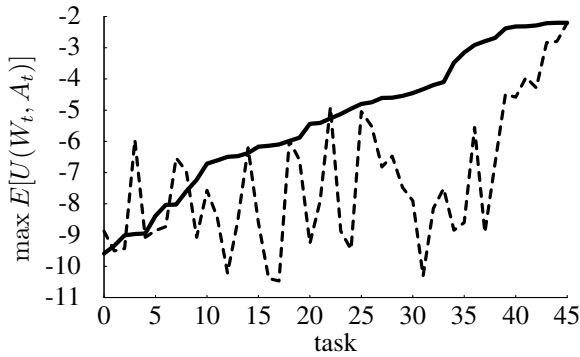


Figure 5: Typical example of utility achievable on each task using the minimal optimal sensor obtained for a specific initial task, denoted by the solid line, ordered from low to high achievable utility given this sensor. The task with the highest order number is the initial task for which the agent was optimized. The dashed line indicates the utility achievable using the action that would be taken for the initial task as the source of information, instead of the sensor input.

an example of a partitioning of the world by such a sensor.

In such a scenario, we can formulate and perform the decoupled evolutionary iterations as given in Alg. 1; a detailed description of step 4 can be found at the end of this paper. The solid line in Fig. 5 shows a typical example of the maximum utility achievable on the full range of tasks given the sensor for the initial task, as found in step 4 of Alg. 1. The most striking observation in the context of our argument, is that there is a group of tasks on which the agent can perform close to the optimum, *despite* the sensor that is used being fully optimized and minimized to provide *only* the information strictly relevant to the initial task.

When we obtain these results for all possible initial tasks, we can construct a directed graph, where each node corresponds to a task, and the heads of the edges indicate for which tasks an agent can still achieve near-optimal performance given the minimally optimal sensor of the predecessor task. Such a graph shows which evolutionary transi-

Algorithm 1 Uncoupled Sensory-Motor Evolution

- 1: Select initial task g
 - 2: Find RI-optimal direct policy $\pi_g(a_t|w_t)$
 - 3: Use IB to find minimal optimal sensor $p(s_t|w_t)$ for this policy
 - 4: Find the optimal policy $\pi_{g'}(a_t|s_t)$ for other tasks given current sensor
 - 5: Determine task g^* with highest performance given sensor, resolving ties by random selection
 - 6: $g \leftarrow g^*$
 - 7: Repeat steps 2–3 for this new task
-

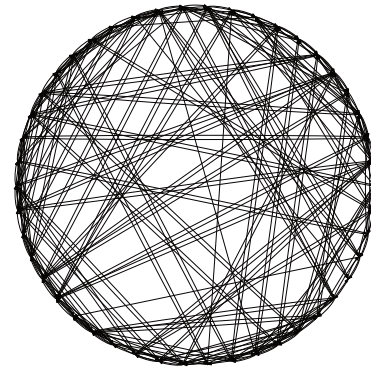


Figure 6: Directed graph showing feasible evolutionary transitions between different tasks under the uncoupled evolution model. Each task is represented by a point on the outer circle (in no particular order), and an arrow from one task to a second indicates that the minimally optimal sensor obtained for the first task allows an expected utility on the second task of no less than 95% than the maximum achievable on that task.

tions are relatively easy to bring about, while at all times moving towards an optimal (local) information-utility trade-off, without the necessity of synchronized adaptation of both sensor and actuation. Figure 6 gives this graph for our example world, connecting only tasks where the achievable performance given the sensor is at least 95% of the maximum performance given the full world state. Even at this threshold, we see that the graph is highly connected, indicating easy and rapid evolution between many tasks. Some further details of this graph are discussed below.

Sensor Evolution for Expanding Behavior Repertoire

In the previous section we have given a model of how evolution could continuously drive an organism from being optimally adapted to one task (niche) to another. These steps can be seen as transitions from a point on the trade-off curve of one task to a point on the curve of another, and these transitions induce a drive to adapt a sensor for the new tasks. In this variant of the model, the complexity of the sensor could even decrease, if this precision is not necessary for the new task. Such an effect is seen in nature for instance in blind Spalax mole rats and cave fish (Fong et al., 1995), that have occupied a niche where eyes are no longer relevant sensors and form an unnecessary burden. In this section we will show how our framework may increase our understanding of how species could be driven towards the other, much more striking extreme we noted in the introduction: where the sensory accuracy is pushed towards the limits of physics.

To do so, we change the interpretation of different reward functions from modeling specific mutually exclusive niches, only one of which an organism can occupy during its life-

Algorithm 2 Sensor Evolution Towards Optimal Precision

-
- 1: Initialize ‘blind’ sensor ($|\mathcal{S}| = 1$)
 - 2: Select initial task g
 - 3: Find RI-optimal direct policy $\pi_g(a_t|w_t)$
 - 4: Use IB to find minimal optimal addition to sensor $p(s'_t|w_t, s_t)$ for this policy
 - 5: Combine the original sensor S_t and the addition S'_t into a new equivalent minimal sensor S_t
 - 6: Find the optimal policy $\pi_{g'}(a_t|s_t)$ for other tasks given current sensor
 - 7: Determine task g^* with highest performance given sensor, resolving ties by random selection
 - 8: $g \leftarrow g^*$
 - 9: Go to step 3 unless all tasks are treated
-

time, to a set of goals that all can be imposed on an organism during its lifetime, drawn from some distribution $p(g)$. In this scenario, the overall performance of the agent is then determined by the expected utility averaged over *all* possible tasks, $E[U(S, A, G)]$. This means that there is a pressure to perform optimally on all tasks, instead of over-fitting on one or a small selection.

We change the iterative decoupled evolutionary model of Alg. 1 at one point in order to fit this scenario: instead of letting the agent’s sensor adapt fully to a new task and by doing so move away from the old task, we let it adapt to incorporate the new task while *preserving* the optimality of its existing repertoire of behavior. This means that, instead of adapting the agent’s sensor to be optimal for the new task in step 3 of Alg. 1, we create an addition to the sensor, S'_t , that is optimized using an information bottleneck such that it captures the relevant information for the new task, *beyond* what is already available in the existing sensor. Formally, this is done by minimizing $I(W_t; S'_t)$ under the constraint that $I(S_t, S'_t; A_t) \stackrel{!}{=} I(W_t; A_t)$. This process can then be repeated, increasing the precision of the sensor at each step, until the agent’s sensor has reached the maximum required precision to allow the agent to achieve all possible tasks optimally. This new iterative model is detailed in Alg. 2, of which step 5 is elaborated in the appendix.

Performing this process in our grid-world scenario, and determining the overall performance of the agent at every iteration, gives the development curve shown in Fig. 7. This curve shows that indeed every adaptation to add a single task to the agent’s repertoire monotonically increases the performance on the full range of tasks, even though at each step its sensor is *only* explicitly optimized to support only a limited range of tasks. The most striking aspect however is how *rapidly* the sensor is driven toward the globally optimal precision: after optimization for only 7 of the total of 46 tasks (less than 20%) the sensor is already precise enough to be able to perform near to optimum globally, with full optimal-

ity possible after only 7 more epochs. Figure 4c shows the goals of the first 14 iterations. Note that the set of goals does not grow out from the first goal, but rather that successive goals can be some distance apart, but also that the final set of goals still only cover a distinct area, which apparently is enough to require a sensor to be accurate enough to reach any possible goal in the world optimally.

Concomitant Sensor Information as a Major Evolutionary Drive

The iterative model that we presented here is able to show that sensory evolution can be driven by the adoption of a novel behavior/niche that is already well supported by the existing sensor, after which the sensor can be optimized for the new (repertoire of) behavior. Our results show that this process can rapidly bring about large evolutionary steps, based on the observation that, even when a sensor may be adapted fully for a single task, it still enables the achievement of different tasks near to optimality, or even fully optimally. An important question is whether this is an artifact of our particular examples or model, or whether this is likely to hold more generally. In other words, are these dynamics generic? We argue that there is indeed a structural aspect of the PA-loop that facilitates adaptation towards novel optima, and that this aspect is reflected directly in the informational structure of the system.

In the information bottleneck paradigm it is known that the amount of information that a bottleneck variable (here: the sensor state) can capture about the source variable (the world state) can be significantly larger than the amount it gives about the relevance variable (the action). Moreover, one can show formally that this inequality must hold for *all* possible combinations of worlds, sensors and policies, by employing the general information theoretic law of data processing inequality (Cover and Thomas, 1991). In our framework this means that $I(W_t; S_t) \geq I(S_t; A_t)$, which we indeed encounter: in our scenarios the first term is between two to three times greater than the second. This observation is important: such a large amount of additional information available in the sensor state greatly increases the chance of a significant overlap with the information relevant for other task.

From this, we arrive at the hypothesis that this *concomitant information*, that comes piggyback with the relevant information in a minimal optimal sensor, is a major factor in enabling sensory-actuation evolution.

To test this, we consider the maximum achievable performance on novel tasks using the sensor, which is likely to carry concomitant information, and compare it to the level achievable when strictly using only the minimum of information relevant to the initial task. This ‘strict’ relevant information is expressed in the final actions selected (Salge and Polani, 2010), so to obtain the latter performance we can alter step 4 of Alg. 1, to instead use the action selected

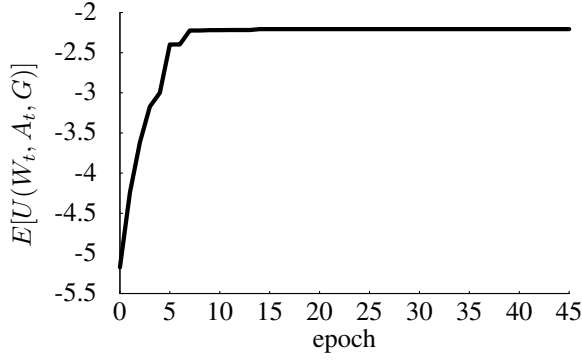


Figure 7: Typical example of the development curve of an agent in the grid-world navigation scenario.

according to the policy $\pi(a_t|w_t)$ as our ‘sensor’. The results of this for our example scenario are depicted by the dashed curve in Fig. 5. They show that for many of the possible novel tasks, using the full sensor enables a significantly higher performance compared to utilizing only the relevant information captured in the policy, as would be predicted from our hypothesis.

Discussion

We have given a general model based on information-theoretical concepts of uncoupled sensor and actuation evolution, and shown how in this model evolutionary jumps between locally minimal optimal sensori-motor trade-offs can be facilitated.

The edges in a transition graph such as Fig. 6 give insight into the ease with which evolution can explore the full space of possibilities. Firstly, we can note that from each point a major subset of the other points can be reached through a limited number of transitions, implying that even a highly specialized species could evolve away into a wide range of completely different niches. Secondly, the fact that from many points not just one, but several points are directly reachable, indicates a possibility for diverging evolutionary pathways. And finally, the graph uncovers the irreversibility of parts of the evolutionary process. This is exhibited by a number of solutions that are only connected unidirectionally, indicating that the optimal sensor for one task is usable for the second, without the optimal sensor for the second supplying enough relevant information for the first task. Further graph-theoretical analysis of this graph, e.g. determining its radius, components, etc., or by integrating a similarity measure between tasks and/or between the minimally optimal sensors for those tasks, may uncover other interesting aspects, however this is outside the scope of the current paper and will be studied later.

The most striking result of the current work is presented in Fig. 7, which shows a strong drive towards optimal sensory precision. The gradient of this curve indicates a significant

pressure to optimize a sensor for novel behavior. This occurs because this not only adapts the agent optimally to that specific novel behavior, but the improvements of the sensor that follow this adaptation turn out to make a significant range of other beneficial behavior feasible as well.

We argue again that the major facilitator of this process is the concomitant information, that is available in a sensor *beyond* that which is purely relevant, *even* in a sensor that is explicitly informationally minimal. Notably, the presence of concomitant information is not an aspect of our specific model, but derives from general basic information-theoretical laws. The fundamentality of this phenomenon leads us to hypothesize that it may not only be one of the major drives in sensor evolution, but that it could also play a large role in the evolution of many other aspects of cognitive systems. For instance, if the concomitant information is relevant to future behavior, it may significantly accelerate the evolution of memory. Taking this concept still further, it may even offer an insight into examples where relevant information happens to be captured by non-sensory systems, driving them to be adapted as useful sensors, as happened with lung-based hearing in amphibians (Hetherington and Lindquist, 1999). Such directions of further exploration of the phenomena could give important insights into evolution and the importance of information therein, and therefore will be the topic of future research.

Appendix: Methodological Details

Policy Optimization for Novel Tasks

A *value-iteration* (Sutton and Barto, 1998) type method is used to find the maximum achievable performance given a fixed sensor mapping $p(s_t|w_t)$. Here, the following is iterated until convergence, starting with a random policy π :

1. Iterate Eq. (1) until convergence w.r.t $U^\pi(w_t, a_t)$
2. Determine $U^\pi(s_t, a_t) = \sum_{w_t} p(w_t|s_t)U^\pi(w_t, a_t)$
3. Set policy to be greedy with respect to the new utility estimate, i.e. $\pi(a_t|s_t) \leftarrow 1/n$ if $U^\pi(s_t, a_t) = \max_{a'_t} U^\pi(s_t, a'_t)$, otherwise $\pi(a_t|s_t) \leftarrow 0$. Here, n is the number of actions having the maximum utility, i.e. $|\{a_t : U^\pi(s_t, a_t) = \max_{a'_t} U^\pi(s_t, a'_t)\}|$.

Finally, perform 1. to find the ultimate maximum performance $E[U^\pi(W_t, A_t)]$ given the final policy and sensor combination.

Due to the partial observability induced by a limited sensor, this process may not converge, but end up in an oscillation between a number of policies. In this case we stop after 1000 iterations and use the best policy in this oscillation. This may not be the global optimum, however this oscillation only occurs for tasks for which a sensor is notably unfitting, and thus does not influence our model, which is only concerned with well fitting tasks.

Sensor Extension and Merging

The bottleneck variables used in Algs. 1 and 2 (i.e. S_t and S'_t) have the same cardinality as the full world state variable, to ensure that there is no structural limitation on how much information they can capture. However, naively combining the existing sensor, S_t , and the addition optimized for a novel task, S'_t , in Alg. 2 leads to an exponential growth of the sensor size. As this makes the model computational unfeasible, and biologically implausible, we construct an equivalent minimal combination as follows (using Bayes' rule):

1. Determine $p(w_t|s_t, s'_t) = \frac{p(s_t|w_t)p(s'_t|w_t)p(w_t)}{p(s_t, s'_t)}$
2. Cluster all combinations s_t, s'_t that give sufficiently similar conditional distributions of W_t (as measured by the Jensen-Shannon divergence (Cover and Thomas, 1991)) into a single new sensor state.

Practically, this results in a sensor with size no larger than that of the alphabet of world states.

References

- Ay, N., Bertschinger, N., Der, R., Gttler, F., and Olbrich, E. (2008). Predictive information and explorative behavior of autonomous robots. *The European Physical Journal B - Condensed Matter and Complex Systems*, 63(3):329–339.
- Baylor, D. A., Lamb, T. D., and Yau, K. W. (1979). Responses of retinal rods to single photons. *The Journal of physiology*, 288:613–634.
- Bullock, T. H. and Diecke, F. P. J. (1956). Properties of an infra-red receptor. *J. Physiol.*, 134:47–87.
- Cover, T. M. and Thomas, J. A. (1991). *Elements of information theory*. Wiley-Interscience.
- Denk, W. and Webb, W. W. (1989). Thermal-noise-limited transduction observed in mechanosensory receptors of the inner ear. *Phys. Rev. Lett.*, 63:207–210.
- Fong, D., Kane, T., and Culver, D. (1995). Vestigialization and loss of nonfunctional characters. *Annual Review of Ecology & Systematics*, 26:249–268.
- Hetherington, T. E. and Lindquist, E. D. (1999). Lung-based hearing in an earless anuran amphibian. *Journal of Comparative Physiology*, 184:395–401.
- Kandel, E., Schwartz, J., and Jessell, T. (2000). *Principles of neural science*. McGraw-Hill, Health Professions Division.
- Klyubin, A., Polani, D., and Nehaniv, C. (2004). Organization of the information flow in the perception-action loop of evolved agents. In *Proceedings of 2004 NASA/DoD Conference on Evolvable Hardware*, pages 177–180. Published by the IEEE Computer Society.
- Kortmann, R., Postma, E., and van den Herik, J. (2001). Evolution of visual resolution constrained by a trade-off. *Artif Life*, 7(2):125–145.
- Lakes-Harlan, R. and Heller, K.-G. (1992). Ultrasound-sensitive ears in a parasitoid fly. *Naturwissenschaften*, 79:224–226.
- Laughlin, S. B., de Ruyter van Steveninck, R. R., and Anderson, J. C. (1998). The metabolic cost of neural information. *Nature neuroscience*, 1(1):36–41.
- Pfeifer, R., Lungarella, M., Sporns, O., and Kuniyoshi, Y. (2007). On the information-theoretic implications of embodiment – principles and methods. In *50 Years of Artificial Intelligence*, volume 4850, pages 76–86. Springer-Verlag.
- Polani, D. (2009). Information: currency of life? *HFSP journal*, 3(5):307–16.
- Polani, D., Nehaniv, C., Martinetz, T., and Kim, J. (2006). Relevant information in optimized persistence vs. progeny strategies. In *Proceedings of Artificial Life X*, pages 337–343.
- Salge, C. and Polani, D. (2010). Digested Information as an Information Theoretic Motivation for Social Interaction. *Journal of Artificial Societies and Social Simulation*, 14(5).
- Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA.
- Tishby, N., Pereira, F. C., and Bialek, W. (1999). The information bottleneck method. In *Proceedings of 37th Annual Allerton Conference on Communication, Control and Computing, Illinois*.
- Tishby, N. and Polani, D. (2011). Information theory of decisions and actions. In *Perception-Action Cycle*, Springer Series in Cognitive and Neural Systems, pages 601–636. Springer New York.
- Touchette, H. and Lloyd, S. (2000). Information-theoretic limits of control. *Physical Review Letters*, 84(6):1156–1159.
- Touchette, H. and Lloyd, S. (2004). Information-theoretic approach to the study of control systems. *Physica A: Statistical Mechanics and its Applications*, 331(1-2):140–172.
- van Dijk, S. G., Polani, D., and Nehaniv, C. L. (2010). What do You Want to do Today? Relevant-Information Bookkeeping in Goal-Oriented Behaviour. In *Proceedings of Artificial Life XII*, pages 176–183. MIT Press.

Task decomposition with neuroevolution in extended predator-prey domain

Ashish Jain¹, Anand Subramoney¹ and Risto Miikulainen¹

¹University of Texas at Austin, Austin, TX 78701
{ajain,anands,risto}@cs.utexas.edu

Abstract

Learning complex behaviour is a difficult task for any artificial agent. Decomposing a task into multiple sub-tasks, learning the sub-tasks separately, and then learning to use them as a whole is a natural way to reduce the dimensionality and complexity of the task function. This approach is demonstrated on a predator agent in the predator-prey-hunter domain. This extended domain has a new agent, a ‘hunter’, that chases the predators. The evading and chasing behaviours are learnt as separate sub-tasks by separate networks using the NEAT neuro-evolution method. A separate network is then evolved to use these networks based on the situation. Task decomposition using this approach performs significantly better in the predator-prey-hunter domain compared to a monolithic network evolved directly on the whole task.

Introduction

Developing complex behavior using machine learning is still a challenging goal for artificial life. At a high level, many such problems have a natural solution – split the large complex task into smaller manageable parts. Solving the parts may be easier than solving the entire problem at once and these smaller solutions then can be combined to give a solution for the entire problem.

This paper presents a neuroevolution approach to such a task decomposition in a predator-prey domain with multiple hunters chasing a predator that is also trying to catch a prey. The predator-prey domain is a well studied problem in machine learning (Benda et al. (1986)). It has also been studied in several variations in the evolutionary context (Luke and Spector (1996), Miller and Cliff (1994), Haynes and Sen (1996), Yannakakis and Hallam (2005), Yong and Miikulainen (2010), Rajagopalan et al. (2011)). Although it is not a complex real-world domain, it is a versatile domain that can be used to illustrate important concepts of problems and approaches.

There are multiple parameters that can be varied in the predator-prey domain including relative speed of the prey with respect to the predator, number of predators, number of prey, the type of the world (continuous, closed toroidal, plane etc.), having separate teams of predators and/or prey,

whether both the predator and prey learn or one has fixed behaviour, etc. Additional goals may also be added to the problem apart from capturing prey. Each of these variations alters the problem significantly and also changes the difficulty of learning the problem significantly. For instance, having multiple predators, and defining the capture method as one or more predators occupying cells adjoining the prey in all directions makes the task cooperative. On the other hand, allowing only one of the predators to capture the prey at a time, and that predator receiving the entire reward for capture of the prey, makes the problem competitive. The prey may also be evolved along with the predator, leading to an arms race between the predators and the prey. The introduction of multiple agents and multiple sub-goals makes the domain quite difficult for a simple network to solve.

Neuro-evolution as a method of training neural networks has been successfully used to solve large complex domains Yao (1999), Stanley et al. (2005), Floreano and Urzelai (2000), Gomez and Miikulainen (1997). Although computationally more intensive than back-propagation, it is less prone to stagnation and more efficient in searching complex landscapes. One of the more successful neuro-evolution techniques is Neuro-evolution of Augmenting Topologies (NEAT) Stanley and Miikulainen (2002). NEAT evolves increasingly complex networks in each generation, starting from a very simple network. NEAT is chosen here because it evolves both the weights and the topology of the network and it tends to find a solution close to the minimal size.

In this paper, the predator-prey-hunter task is decomposed into predator-prey and predator-hunter tasks. Networks are trained using NEAT on each of these tasks. These sub-networks are combined using another selection network that is also trained using NEAT. This selection network chooses between the sub-networks given the positions of the hunters and prey in the overall task. Such a decomposed and hierarchical approach is shown to perform much better than training a single monolithic network for the overall task. It is also shown that with increasing complexity of the domain (with more hunters), this hierarchical approach outperforms the monolithic network by an increasing magnitude.

Related work on task decomposition is first discussed. A brief outline is then given of the predator-prey domain followed by a description of the approach to solving the extended predator-prey domain with multiple sub-goals using task decomposition. The experiments are described in detail and the paper concludes with the discussion and future work.

Related Work

Task decomposition has been studied before in several other domains using different training and combination methods.

Lee (1999), studied the task of finding a box in an enclosure and pushing it towards a light source by a robot, decomposing it into separate subtasks of finding the box, positioning the robot, and pushing the box in a straight line. Separate controller circuits were evolved in simulation for each of the sub-tasks, one at a time, using Genetic Programming (GP). Then higher level controller circuits were then evolved to select the appropriate sub-task controller based on the sensory inputs. Such a decomposition of the overall task into separate subtasks performed better than evolving a monolithic controller circuit. The current paper follows a similar approach but evolves neural networks using NEAT instead of controller circuits.

On the soccer-keepaway task, Whiteson et al. (2005) evolved an agent for playing keep-away. Keep-away is a subdomain in robo-soccer where one team of agents, the *keepers*, try to keep the ball away from the other team, the *takers*, within a given fixed region. The subtasks were: intercepting a pass, passing the ball to a team mate, evaluating if passing a ball to a particular team mate is viable, and moving to a good position for intercepting the ball. The agent was trained for each of these sub-tasks separately, and then all these networks were combined, using decision tree in one case, and a combiner network in the other case. Their performance was compared to the case where a single network was evolved for all four tasks simultaneously. The task decomposition gave significantly better results than having a single monolithic network. Apart from the overall performance, there were some interesting behaviours observed in the modular network that was not present in the monolithic network. In particular, the agent learnt to approach the ball from the direction opposite to that in which it was going to kick the ball, since “kicking” the ball in this domain was actually coming in contact with the ball at the right velocity. The agent actually learnt that it was inefficient to first approach the ball from an arbitrary direction, and then move to the right position to kick the ball. Task decomposition gave good results only when a fixed decision tree was used to combine the subnetworks. Evolving the combiner network didn’t perform as well as the fixed decision tree. The goal of the current paper is to show that a proper combination of the subtasks enables the combiner network to be learned as well.

There has also been some work done on learning tasks incrementally (Gomez and Miikkulainen (1997) being one of them) – starting with a simple task, and slowly increasing the task difficulty as the network learns. This approach is different from the task decomposition addressed in this paper in that the task remains the same, and just a few parameters of the task are varied to make it more difficult. For instance, in the predator-prey domain, the speed of the prey is increased slowly. In contrast, the task decomposition approach in the current paper divides the task into specific sub-tasks and later combines them.

In Yong and Miikkulainen (2010), multi-agent ESP (Enforced Sup-Populations) was used to coevolve multiple networks for each set of inputs for a predator-prey task, and it was shown that this coevolved network performs better than a monolithic network when there were multiple predators and prey involved. This work was extended in Rajagopalan et al. (2011) to domains with different types of prey, and with individual and shared fitness, where cooperation between the agents was seen to evolve. Multi-agent ESP decomposes the overall network in terms of the inputs automatically, but cannot be directly applied to arbitrary task decomposition. The current paper develops a mechanism to decompose networks for arbitrary (manually specified) task decomposition. Currently the NEAT neuroevolution method is used, although in the future, multi-agent ESP could be modified to work for arbitrary task decomposition.

The Extended Predator-Prey Domain

A toroidal grid world of size 10×10 with one predator, one prey and multiple hunters is used. This is illustrated in Figure 1, which has four hunters (filled blue circles), one predator (red square) and one prey (black circle). The agent being evolved is the predator. The goal of the predator is to capture the prey in as few time steps as possible without being caught by the hunter(s) in the process.

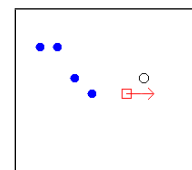


Figure 1: Illustration of the extended predator-prey domain. The four filled blue circles are the hunters chasing the predator, which is indicated by the red open square. The black open circle is the prey being chased by the predator.

“Capture” of the prey is defined as the predator occupying the same cell as the prey. Likewise, capture of the predator is defined as a hunter occupying the same cell as the predator. The behaviour of the prey and the hunters are fixed. The prey always moves away from the predators with a fixed move probability, while the hunter moves towards the preda-

tor with a fixed move probability. The prey and the hunters move slightly slower than the predator, their move probabilities being 0.8 (and hence their speed is 0.8 times the speed of the predator).

If the predator is caught by the hunters, it receives a large negative reward equal to -10 times the number of remaining steps in the episode. And hence the predator agents have to learn to stay away from the hunters, apart from chasing and capturing the prey. On capturing the prey, the predator receives a large positive reward of 10 times the number of remaining steps in the episode. If the predator neither catches the prey nor is caught by the hunter at the end of 100 steps, the episode ends and the predator receives a small positive reward equal to the difference of its distance from the hunter and its distance from the prey. This means that it receives a larger reward if it is farther away from the hunter and closer to the prey.

In this extended domain, the two tasks that the predator has to do – running away from the hunters and chasing the prey are not completely independent. If the predator were to blindly chase the prey (or blindly evade the hunter), it will not be successful. It would keep getting caught by the hunter (or not catch the prey at all), since the hunter is programmed to always chase the predator (and the prey to always run away from the predator). A successful strategy would involve doing both tasks simultaneously as much as possible, and if not, run away from the hunter, since the reward on capture by the hunter is negative. The tasks are not very tightly coupled either, in the sense that the predator does not always have to do both simultaneously to be successful. A strategy of alternation between the tasks would also work. The primary reason such a task was chosen was that (1) the behaviours for each task is easily identifiable and well defined, and (2) the tasks are neither too tightly nor too loosely coupled. It provides, in the authors' opinion, a good balance of behaviours similar to those many animals exhibit in the real world.

Method

The task of the predator agent is decomposed into two parts – capturing the prey, and avoiding the hunter. The predator agent is trained separately on each of these subtasks i.e. the agent is first trained in an environment with only one prey and no hunters where it learns to capture prey successfully. Then the agent is trained in an environment with only one hunter and no prey, where it learns to avoid the hunter successfully.

A selection network is then evolved in the presence of one prey and multiple hunters. This selection network chooses between the outputs of the one modular prey chasing network (which gets the relative position of the prey as input) and n modular hunter evading networks (each of which gets the relative position of one hunter as input). These n hunter evading networks are copies of the hunter evading network

evolved in the subtask. The selection network sees the entire domain, i.e. the positions of the prey and all the hunters. The task of the selection network is to decide which agent it wishes to chase or avoid at any given time step given this information. In practice only one of two networks (the prey chasing or hunter avoiding network) has to be activated, with the relative positions of the selected agent as the input. Since the selection network selects only one task at a time, it would seem that it might have trouble doing both the evading and chasing simultaneously. But, as will be seen later, the selection network is able to switch between multiple tasks fast enough to accomplish both the tasks simultaneously, and it does it surprisingly effectively. The results of this selection network are compared with a monolithic network – a single network that is evolved to solve the entire domain (without any modularity).

Experiments

The experiments were conducted on a 10×10 toroidal grid. Since the normalized relative x and y positions of the prey and hunters are provided to the predator the effect of increasing the grid size is not significant.

A total of six experiments each were conducted for both the monolithic and the selection network. For each experiment the predator network was evolved for 200 generations. Each experiment was conducted 30 times and the results were averaged. Two hundred generations was chosen because while running initial experiments for 1000 generations it was seen that the fitness of the monolithic and selection network did not change much after 200 generations. NEAT neuroevolution was restricted to feed forward networks for simplicity, since memory was not strictly required to solve this task. The number of hunters was varied from 0 to 5 in six separate experiments. The prey chasing modular network was evolved for 1000 generations with only one prey and no hunter in the domain. Likewise, the hunter evading modular network was evolved for 1000 generations with only one hunter and no prey in the domain. At the end of each generation the champion fitness i.e. the fitness of the best performing network in that generation, was saved.

The same chasing and evading networks were used in all six experiments, i.e. only the selection network was evolved in each of them.

Results

The champion fitness for experiments conducted with 1, 3 and 5 hunters are shown in figures 3, 4, and 5, respectively. The results are summarized in table 1. Figure 2 shows the mean champion fitness as the difficulty of the task i.e. the number of hunters increases.

As can be seen from figure 2, the task decomposition approach performs better than the monolithic approach in every experiment with any hunters in the world. Further, as the difficulty of the task increases (number of hunters > 3),

Champion Fitness			
Number of Hunters	Monolithic	Selection	Percentage Improvement
0	822.66	832.00	1.13
1	703.65	752.50	6.94
2	503.68	704.46	39.86
3	204.02	649.40	218.29
4	-138.69	278.80	301.01
5	-79.41	289.96	465.14

Table 1: Champion fitness of generation 200 (averaged over 30 experiments). Notice that for domains with 4 and 5 hunters the monolithic network has very low fitness and is unable to catch the prey at all on average whereas the selection network is still able to catch the prey.

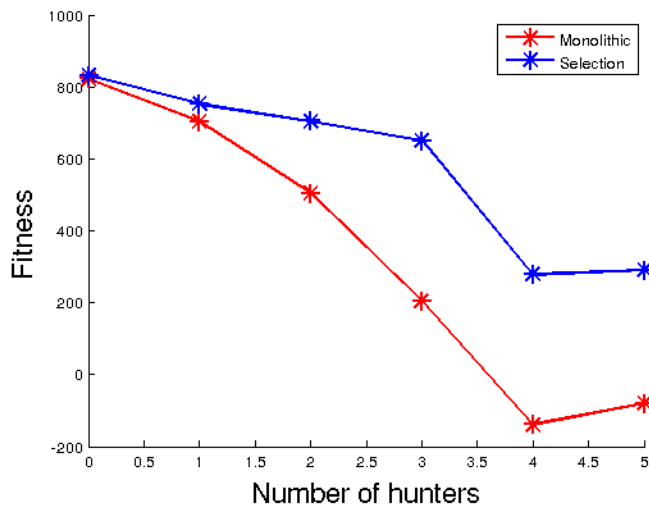


Figure 2: Average Champion fitness of Monolithic network (red) and Selection network (blue) as the number of hunters increases. The average champion fitness has been computed using champions of generation 200 averaged over 30 experiments.

the selection network does increasingly better, relative to the monolithic network. Inspection of its behavior suggests that with that many hunters, it cannot balance the two tasks, but focuses on mostly surviving or escaping from the multiple hunters. On the other hand, the selection network deals with the increase in the difficulty of the task gracefully. It is able to catch the prey even with more than three hunters although it takes more time.

Table 2 shows how the number of hidden neurons of the champion networks varies as the difficulty of the task increases. Each hidden neuron increases the number of parameters, which can be detrimental in finding the optimal solution. The selection network searches for a solution in a much smaller space compared to the monolithic network, making it easier to find good solutions, which is indeed the main benefit of task decomposition.

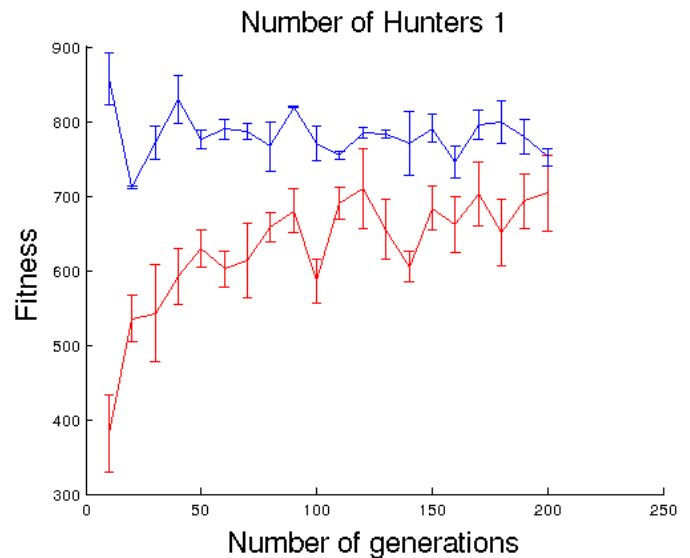


Figure 3: Champion fitness of Monolithic network (red) and Selection network (blue) with only one hunter and one prey. The results have been averaged over 30 experiments. The error bars represent the standard deviation.

Behavior The monolithic network was strongly affected by hunter movements. Even though for the one hunter and two hunter case, the monolithic network was overall focused on the prey, it reacted sharply to the hunter movements. As a result, it changed tracks quite often. As a result, it lost time, and consequently scored lower in fitness. Its reaction to the hunters became dominant in the 3, 4, and 5 hunter case. The monolithic network lost a lot of opportunities to capture the prey even when the prey was close by (figure 6), because the dominant behaviour it learnt was to avoid the hunters.

On the other hand, the selection network was not easily perturbed by the hunters (figure 9). Furthermore, the selection network made decisions taking into account the positions of more than one agent. Note that the selection network only decides which modular network to use, and the module decides how to move. Figures 7 and 8 show how the

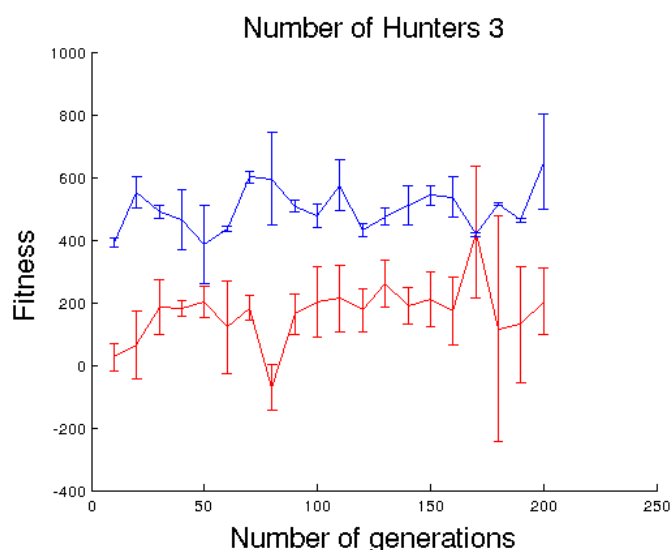


Figure 4: Champion fitness of Monolithic network (red) and Selection network (blue) with three hunters and one prey. The results have been averaged over 30 experiments. The error bars represent the standard deviation.

Number of Hidden Neurons		
Number of Hunters	Monolithic	Selection
1	136	88
2	182	96
3	120	101
4	153	107
5	163	126

Table 2: Number of hidden neurons of the champion network of generation 1000 for the monolithic network and the selection network.

predator takes into account the prey position as well as the hunter position in order to decide its next move.

Figure 10 shows a case where the selection network chooses between the network corresponding to chasing the prey and the one corresponding to evading the hunter, depending on their positions. In figure 10(a) the predator is chasing the prey, but when the hunter gets too close, it switches to evading it as seen in figure 10(b). In the next time step, it goes back to chasing the prey as seen in figure 10(c). The selection network was observed to predominantly choose the network corresponding to chasing the prey, but occasionally selected the network corresponding to evading the hunter in case the hunter got too close while the prey was far. The selection network was also observed to choose the network corresponding to evading the hunter for the first few steps at the beginning of each episode. It should be noted that most of the time, chasing the prey also gets the predator away from the hunters, and the few times it doesn't, the

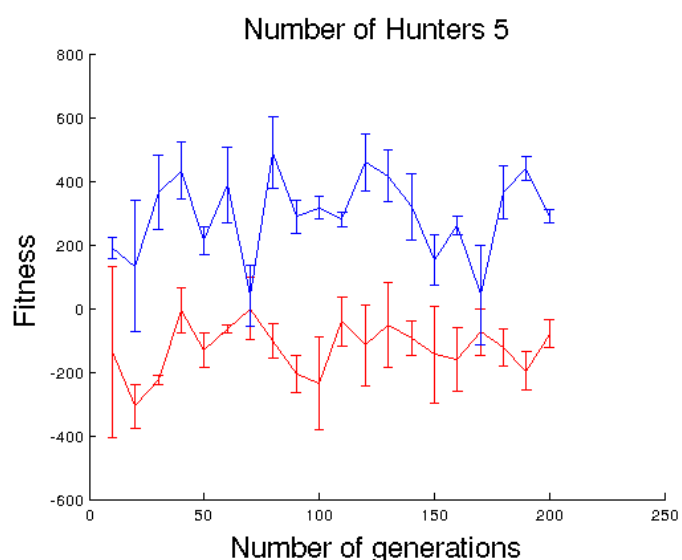


Figure 5: Champion fitness of Monolithic network (red) and Selection network (blue) with five hunters and one prey. The results have been averaged over 30 experiments. The error bars represent the standard deviation.

predator explicitly evades the hunter.

The selection network was sometimes caught while focusing on the prey and ignoring the hunters. This is attributed to the small randomness present in the movement of the prey and hunters, as a result of which it could sometimes catch the prey while ignoring the hunters, and sometimes it was caught while exhibiting the same behavior. Overall, however, such risk taking was effective, which may be why it evolved.

Discussion and Future Work

In this paper, learning a complex task using task decomposition was shown to perform significantly better than using a monolithic network. The task decomposition performs better the more complex the domain is. Note that in the current approach task decomposition has to be done with human input. The way the task is decomposed may not be obvious or unique for most domains. Further, there may exist domains where the tasks are too tightly coupled to be amenable to task decomposition. However, when it is applicable, the results in this paper show that task decomposition is a powerful approach.

There are also multiple avenues for extensions of this work. Broadly, these can be classified as (1) changing the methods of combining subtasks, (2) changing the type of networks itself, (3) giving different types of input to the networks and (4) applying the approach to more complex domains. Apart from these four broad categories, the two other major possible extensions are co-evolving the networks and automating task decomposition. These extensions are de-

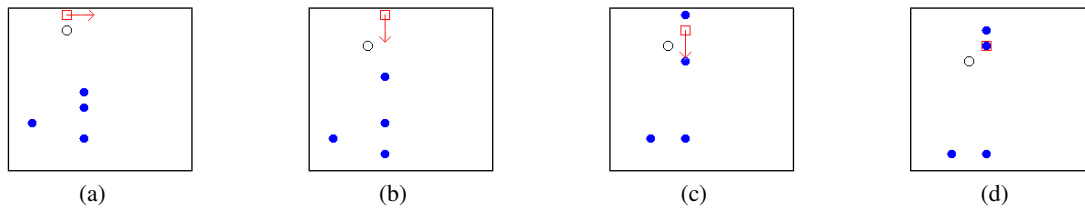


Figure 6: Monolithic network is strongly affected by hunter movements. As can be seen in (a) the predator is very close to the prey. However in (b) it reacts sharply to the hunters especially the one at the bottom (which is closest to it in the toroidal world). As a result it loses sight of the prey, and eventually gets caught as can be seen in (d).



Figure 7: The selection network is able to make decisions taking into account more than one agent. As can be seen in (a) the predator has agents to its right and the prey to its left. Instead of moving forward, it moves down, as shown in (b) thereby allowing it to both evade the hunters and come closer to the prey at the same time. Although it might seem that the predator only tries to minimize its distance with respect to the prey, it has to routinely avoid the hunters in order to avoid getting caught to ensure high fitness scores.

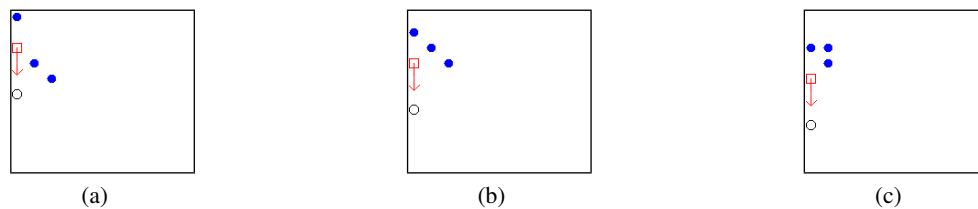


Figure 8: Here we see another instance where the selection network is able to make decisions taking into account more than one agent. As can be seen in (a), the predator is flanked by a hunter on the top and to its right. However, instead of moving left in order to maximize the distance from the hunters, it moves down to also simultaneously try to reduce its distance from the prey. Note that the hunters to the right of the predator move up, as in (b) to minimize the distance with respect to the previous position of the predator.

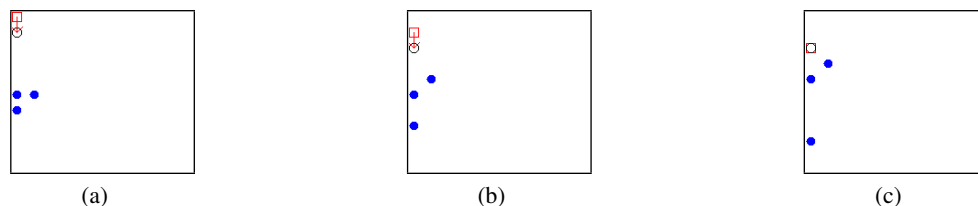


Figure 9: Unlike the monolithic network shown in figure 6 the selection network is not easily perturbed by hunter actions. The predator is chasing the prey in (a). As the hunters get closer in (b), the predator continues to chase the prey, unperturbed, and catches it in (c)

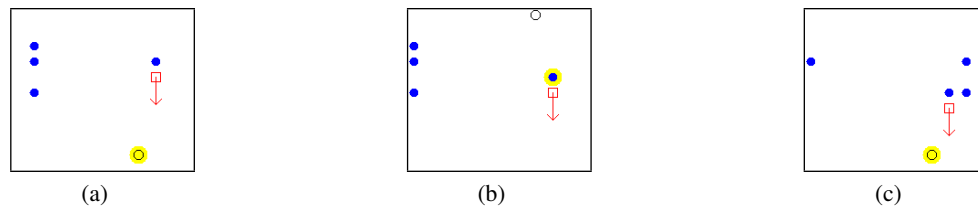


Figure 10: Illustration of switching behaviour between hunter and prey in three consecutive steps. The agent corresponding to the selected network is highlighted yellow. The predator is chasing the prey in (a). As the hunter gets closer in (b), the predator starts evading the hunter for one time step, and goes back to chasing the prey in (c)

scribed briefly below.

In the approach described in this paper, the selection network takes inputs from all the agents in the domain and selects between the various sub-networks. Intuitively, this selection might seem limiting since the network is restricted to selecting just one of the sub-networks in each time step. To allow for more complex behaviour that is a combination of the behaviours suggested by the sub-networks, the combiner network could combine the outputs from all the sub-networks instead of selecting only one. Even using just the selection network, a hierarchy of selection networks could be developed, each one only selecting between two sub-networks. This approach would allow for a more fine-grained control of decomposition and task assignment among the networks.

Given the domain used in this paper, if the predator could keep track of the number of time steps remaining before the end of the episode, it should be able to select a better strategy. For example, if the predator knew that the time remaining is not sufficient to chase down any prey, it could concentrate on avoiding the hunters and not risk getting caught. This information could either be given to the predator as another input, or more generally, recurrent networks could be used.

Co-evolving the sub-networks and the combiner/selection network simultaneously is a promising avenue for extending this work. Rather than evolving sub-networks that only do the task optimally, a sub-network that cooperates well with the other sub-networks and the combiner network would be evolved. This approach would reduce the cases where the combiner/selection network would have to make sub-optimal choices.

There are various ways in which the domain itself might be extended for more complex tasks that might be able to take more advantage of the sub-task decomposition. For instance, it would be interesting to have teams of predators that need to cooperate to achieve the goal. Multiple subtasks that have dependencies on each other, and require a hierarchy of sub-task networks would also be an important step towards simulating complex behavior.

Developing a method to partially or completely automate the task decomposition would help reduce the human input

that is required right now to specify the sub-tasks. It would also help us understand which tasks are amenable to task decomposition and how decomposition contributes to complex behavior.

Conclusion

In this paper, an approach was developed for task decomposition in the neuroevolution framework. This approach is successfully demonstrated on the predator-prey-hunter domain, an extension of the predator-prey domain where there are additional agents (hunters) that can hunt the predators. This approach scales well as the difficulty of the task increases, and consistently performs better and more robustly than the network evolved over the whole task directly. This approach can be seen as a stepping stone to methods that discover task decomposition automatically, thus leading to development of complex general behavior in artificial agents.

References

- Benda, M., Jagannathan, V., and Dodhiawala, R. (1986). On optimal cooperation of knowledge sources – an empirical investigation. Technical Report BCS-G2010-28, Boeing Advanced Technology Center, Boeing Computing Services, Seattle, WA, USA.
- Floreano, D. and Urzelai, J. (2000). Evolutionary robots with on-line self-organization and behavioral fitness. *Neural Networks*, pages 431–443.
- Gomez, F. and Miikkulainen, R. (1997). Incremental evolution of complex general behavior. *Adaptive Behavior*, pages 317–342.
- Haynes, T. and Sen, S. (1996). *Evolving behavioral strategies in predators and prey*, volume 1042 of *Lecture Notes in Computer Science*, pages 113–126. Springer Berlin / Heidelberg.
- Lee, W. (1999). Evolving complex robot behaviors. *Information Sciences*, 121(1-2):1–25.
- Luke, S. and Spector, L. (1996). Evolving teamwork and coordination with genetic programming. In *Proceedings of the First Annual Conference on Genetic Programming*, pages 150–156, Cambridge, MA, USA. MIT Press.
- Miller, G. and Cliff, D. (1994). *Co-evolution of pursuit and evasion I: Biological and game-theoretic foundations*. Brighton: School of Cognitive and Computing Sciences, University of Sussex.

- Rajagopalan, P., Rawal, A., Miikkulainen, R., Wiseman, M. A., and Holekamp, K. E. (2011). The role of reward structure, coordination mechanism and net return in the evolution of cooperation. In *Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG 2011)*, Seoul, South Korea.
- Stanley, K. O., Bryant, B. D., and Miikkulainen, R. (2005). Real-time neuroevolution in the nero video game. *IEEE Transactions on Evolutionary Computation*, pages 653–668.
- Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.
- Whiteson, S., Kohl, N., Miikkulainen, R., and Stone, P. (2005). Evolving soccer keepaway players through task decomposition. *Machine Learning*, 59(1-2):5–30.
- Yannakakis, G. N. and Hallam, J. (2005). *AI in Computer Games: Generating Interesting Interactive Opponents by the use of Evolutionary Computation*. PhD thesis, University of Edinburgh. College of Science and Engineering. School of Informatics.
- Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423 –1447.
- Yong, C. H. and Miikkulainen, R. (2010). Coevolution of role-based cooperation in multi-agent systems. *IEEE Transactions on Autonomous Mental Development*, 1:170–186.

Brainless Bodies: Controlling the Development and Behavior of Multicellular Animats by Gene Regulation and Diffusive Signals

M. Joachimczak¹, T. Kowaliw², R. Doursat^{2,3}, and B. Wróbel^{1,4,5}

¹Systems Modeling Laboratory, IO PAN, Sopot, Poland

²Institut des Systèmes Complexes Paris Ile-de-France (ISC-PIF), CNRS, Paris, France

³Research Group in Biomimetics (GEB), Universidad de Málaga, Spain

⁴Evolutionary Systems Laboratory, Uniwersytet im. Adama Mickiewicza, Poznań, Poland

⁵Institut für Neuroinformatik, Universität & ETH Zürich, Switzerland

Abstract

We present a model of parallel co-evolution of development and motion control in soft-bodied, multicellular animats without neural networks. Development is guided by an artificial gene regulatory network (GRN), with real-valued expression levels, contained in every cell. Embryos develop within a simulated physics environment and are converted into animat structures by connecting neighboring cells through elastic springs. Outer cells, which form the external envelope, are affected by drag forces in a fluid-like environment. Both the developmental program and locomotion controller are encoded into a single genomic sequence, which consists of regulatory regions and genes expressed into transcription factors and morphogens. We apply a genetic algorithm to evolve individuals able to swim in the simulated fluid, where the fitness depends on distance traveled during the evaluation phase. We obtain various emergent morphologies and types of locomotion, some of them showing the use of rudimentary appendages. An analysis of the selected evolved controllers is provided.

Introduction

The *raison d'être* of the nervous systems is to allow for controllable and adaptable movement, but adaptive locomotive behavior exists in the absence of neurons as well. For example, there is evidence that the movement of the multicellular body of certain slime molds, such as *Dictyostelium* (“social amoeba”), results from a difference in activity between the anterior and posterior cells (Bonner, 2008). *Dictyostelium* can respond to minute variations of light, temperature, and concentrations of ammonia and oxygen. In many cases it is known that these stimuli affect the relative location of so-called “organizer cells”, which release a diffusive chemical signal, the same signal used during the aggregation of single cells into the body (reviewed in Kessin, 2001). The relative location of these organizers controls the activity level of the cells across the body, which in turn controls the direction of motion. This impressive capacity of *Dictyostelium* for effective and reactive behavior occurs without any nerve cells.

Dictyostelium is one of the most important “model organisms” in biology for the study of development because its structure is simple and the number of cell types limited. The

assumption that knowledge about complex biological systems can be gained by first studying simpler organisms has proven tremendously successful. We share this view and, in the present work, propose that in order to study body-brain co-development more effectively, it is helpful to consider the basic case of a body devoid of any nervous system. Our approach is related to the investigation of minimal sets of behaviors that can still exhibit interesting “cognitive” abilities (*minimal cognition*; Beer, 1996). In this context, animats capable of executing non-trivial tasks are generated and tested on some cognitive challenge. For example, Dale and Husbands (2009) describe a 1D animat that can perform shape discrimination with limited memory, using only a reaction-diffusion system. Such systems are known to model many developmental processes (Yamada et al., 2007; Lefèvre and Mangin, 2010), hence this choice is consistent with the view that regulation of development and regulation of behavior have mechanisms in common.

The present work brings several new aspects to the discussion of the relations between behavior and development, minimal cognition, and brain-body co-evolution. First, we achieve an important step towards minimal cognition, namely the coordinated behavior of multiple cells, based on a biologically plausible model of gene regulatory networks (GRNs). Second, we utilize the same instance of GRN for both developmental and behavioral control. Our experiments rely on a modeling and simulation platform called GReaNs (for *Genetic Regulatory evolving artificial Networks*), which is dedicated to the study of GRN evolution and evolutionary development based on a linear genomic representation of GRNs. Two of us (Joachimczak and Wróbel, 2011) have shown previously that GReaNs was successful at evolving asymmetrical multicellular structures displaying asymmetrical patterning. We then applied the same model of GRNs to signal processing (Joachimczak and Wróbel, 2010b) and to directing the motion of unicellular animats (Joachimczak and Wróbel, 2010a). In the present work, we rely on another recent extension of GReaNs (Joachimczak and Wróbel, 2012) to model soft-bodied multicellular animats in motion.

On the spectrum of available developmental and generative systems, the GReaNs platform belongs to a relatively small family of models that attempt to retain some degree of “biological realism” (e.g., among others, Mjolsness et al., 1991; Hogeweg, 2000; Salazar-Ciudad and Jernvall, 2002; Doursat, 2008). From the viewpoint of artificial life, these models belong to the “cell chemistry” approaches identified by Stanley and Miikkulainen (2003) in their taxonomic review of artificial embryogeny research. They all attempt to combine the essential chemical and physical principles of both genetic regulation *and* cellular mechanics, and to form fine-grained agent-based modeling rules based on these principles. In such models, the final shape and behavior of an organism are the result of complex interactions taking place at several scales of abstraction. Generally, at the smallest scale, each cell contains a genome that codes for gene products and regulatory sites, and whose interactions (based on sequence-matching in GReaNs) can be mapped to a GRN. On a mesoscopic level, the continuous, dynamic update of product concentrations in the cells leads to various types of cell behavior, such as division and differentiation, as products in the genome build up or degrade over time. Finally, the macroscopic shape and action of the organism emerges from the physical interactions between neighboring cells, which move in space during growth and motion.

In this study, we introduce in the GReaNs model the possibility that global patterns of cell activity—themselves the product of interactions between controller cells, the physical structure of the individual, and the properties of the simulated environment—give rise to the movement of developed multicellular bodies. We show that the control and coordination of this movement do not require an artificial nervous system, but can merely be achieved by decentralized GRN activity in every cell and signal diffusion.

A model of development, behavior and evolution of soft-bodied animats

Genome and GRN

The integrated model of genome, GRN, development and evolution presented in this paper is essentially the same as our recent extension of GReaNs that modeled soft-bodied multicellular animats in motion (Joachimczak and Wróbel, 2012). For the sake of completeness, however, we provide here a full description of the model. The main difference is that, in the experiments shown here, the GRN continues to function during animat movement, while in the previous version the GRN dynamics stopped at the end of development and its final outputs specified the oscillatory behavior of the cells.

A genome in GReaNs is composed of genetic modules or “elements”, which are ordered sets of numbers and belong to three different classes (Fig. 1): G elements code for regulatory products/factors, an abstraction of the biological

transcription factors and diffusive products; P elements are regulatory regions that control (promote or repress) the expression of G elements; and S elements are used as inputs into, and outputs from, the network.

A linear genome is parsed sequentially to build a GRN in which nodes correspond to *regulatory units*. A regulatory unit is a contiguous series of P elements followed by a contiguous series of G elements in the genome. The factors coded by G elements belonging to one unit have the same concentration. As for S elements, they are each mapped to a separate node: when the S element corresponds to an input—to a node with only one regulatory factor (an input factor), when it corresponds to an output—to a node with one regulatory region and one product (an output factor). Output factors determine the actions performed by the cell but do not have affinity to regulatory regions. Products coded by G elements can have affinity to P elements or regulatory regions in output nodes. Factors coded by input S elements can only have affinity to P elements.

The internal structure of each genetic element is composed of several fields (Fig. 1): a *type* field, which specifies the exact type of the element (subtype of G, P or S); a *sign* field; and *coordinate* fields which specify a point in \mathbb{R}^N space (here $N = 2$). The affinity between a regulatory factor and a regulatory region is a decreasing exponential function of the Euclidean distance between their 2D points (weight reaches maximum 10 when points overlap), with a cutoff value to prevent full connectivity (weight is 0 when points are too far apart). The sign of the weight (and thus if it contributes to inhibition or excitation) is determined by multiplying the *sign* fields of the respective elements. Since one regulatory unit of the GRN can be composed of multiple P and G elements, any two nodes in the graph can be connected together through multiple edges. There is no limit on the size of the GRN (number of nodes) in GReaNs.

The concentrations of factors are updated in discrete time steps. First, the activation level of each regulatory region of a node is defined as the weighted sum of the concentrations of all factors (possibly from other units) that have a non-zero affinity to it. If the node corresponds to a regulatory unit, the activation of all P elements of a unit is summed. The rate at which the concentration of factors of a node change is determined using the following update rule:

$$\Delta L = \left(\tanh \frac{A}{2} - L \right) \Delta t \quad (1)$$

where Δt (the integration time step) determines how fast the factors accumulate or degrade in relation to the simulation time step (the value 0.05 is used in this paper), L is the current concentration of the factors in the node (if there is more than one, all have the same concentration), restricted to the interval $[0, 1]$, and A is the summed activation of all P elements in the unit (the effect of a product on a promoter is calculated by multiplying the product’s concentration by the weight).

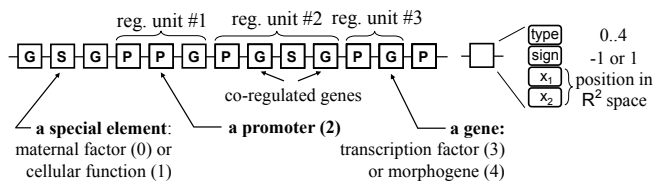


Figure 1: Genome and structure of a single genetic element. Each element consists of a *type* field, which specifies the class of the element (G, P or S), a *sign* field, and a sequence of N abstract *coordinates* in \mathbb{R}^N space ($N = 2$ here), which determine its affinity to other elements.

The S elements of the genome are used to code for GRN inputs and outputs, which provide to a cell certain external signals and the ability to perform certain actions. The concentration of input factors is determined outside of the cell and they diffuse in the physical space of the developmental process (here, in 2D). They can be seen as playing the role of “maternal morphogens”. We used here four different input factors, three of which were produced by sources at specific locations. The fourth factor had a uniform concentration of 1 across the entire space.

Outputs correspond here to six possible cellular actions. The first four actions—cell division, change in cell orientation (rotation to the left and to the right), and change in cell size—affect only development, while the other two actions—coding for cell contraction and expansion—affect only the physical motion of the multicellular animat.

Developmental process

The developmental process starts from a single cell. Each cell contains a copy of the genome, which encodes the GRN and whose activity controls the cell’s developmental behavior. This behavior comprises mechanical rules and chemical rules, which are coupled and influence each other.

Mechanical rules: Cells occupy real-valued positions in 2D space (Fig. 2). An embryo develops in a simulated fluid-like environment, in which cells behave as soft (non-rigid) physical objects. The overall structure of the embryo is maintained by elastic forces between nearest-neighbor cells. Forces are repulsive when cells are too close and attractive otherwise, reaching an optimal distance at equilibrium. After each division of a mother cell, the two daughter cells partially overlap (see rotation action below), so they immediately repel each other.

Chemical rules: Exogenous maternal morphogens located in the environment allow differentiation based on cells’ location in space. Cells also produce endogenous diffusive factors that affect morphogenesis (morphogens). In the simplified, grid-less diffusion model used here, the concentration of these regulatory factors in a cell at a given location is a function of the distance from the source and (for endogenous factors) the historical concentration in the source cells.

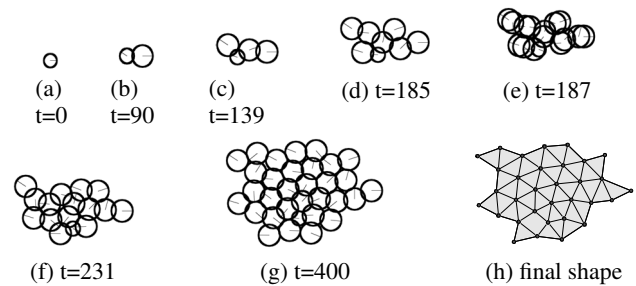


Figure 2: Example of the developmental mechanics. Cells are represented as circles. In (e), cells have just divided but elastic forces have not yet pushed them apart. This was achieved in (f). (h) shows the final structure after cells were connected with springs, see Fig. 4a for the same animat in motion.

Mechanical-chemical coupling: We describe the first four output functions mentioned above. *Cell division* is triggered when the concentration buildup of a specific “division factor” (coded by one of the S elements) reaches a threshold of 0.9. Should this element become disconnected from the GRN (due to mutation) or lost (due to deletion), the individual would consist of a single cell and have zero fitness. The division is asymmetric: a new “daughter” cell is formed from a given “mother” cell. In this paper, there is no asymmetry in the distribution of gene products (the daughter inherits all the concentrations from the mother), but rather in the cell’s size and orientation angle. This angle is an abstraction of the cell’s polarization axis and/or cleavage plane and determines where the daughter cell is placed with respect to the mother. The orientation of the mother cell remains the same after division, while *cell rotation* factors change the daughter cell’s angle proportionally to their concentration. A “right rotation factor” causes an increase of the angle, while a “left rotation factor” causes its decrease (a $\pm 2\pi$ rotation corresponds to the maximum concentration 1 of the right/left factor). Finally, *size increase* determines the radius of the daughter cell at division, which may be up to 1.5 times the default radius when the concentration of the corresponding “size factor” is at the maximum of 1.

Final structure

The developmental phase is followed by a transformation of the obtained morphology into the actual structure of the animat (Fig. 3). In principle, this transformation restricts the set of evolvable structures, but it is also a way to keep the evolutionary search focused, provided that such restriction is still able to produce individuals that are diverse and relevant to the challenge at hand.

The first step of the transformation process consists of outlining a tight, but not necessarily convex, hull that encloses all the cells. This requires identifying the “outer” cells and connecting the centers of adjacent cells with edges,

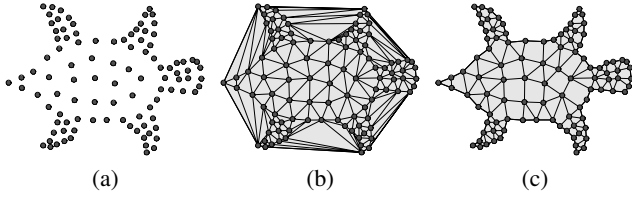


Figure 3: Algorithmic transformation of a set of points into an animat structure: (a) cell centers at the end of the developmental phase, (b) Delaunay triangulation of the set, (c) Gabriel graph of the set (final structure).

while preserving “concave” regions. The resulting hull corresponds to the external surface or “skin” of the animat’s body, which in a simulated fluid-like environment is the only source of drag forces. In a second step, the animat’s internal structure is completed by connecting all the remaining neighboring cells through elastic edges modeled as damped springs. This structural graph is calculated on the basis of cells’ centers only. Cells’ radii affect the final structure only implicitly, by determining the equilibrium positions of the cells during development.

To calculate connectivity, we use a particular notion of spatial proximity defined by the *Gabriel graph* (Gabriel and Sokal, 1969), which is different from nearest neighbors: any two points will be connected by an edge if and only if there are no other points inside the circle whose diameter is that edge. The Gabriel graph is a convenient way to obtain non-convex hulls: it is non-parameterized, scale invariant, and relatively straightforward to compute. Because it is a sub-graph of the Delaunay triangulation, it can be derived from the latter in linear time by removing all the edges that do not fulfill the above proximity criterion.

Motion generation

The final structure of the animat defines a soft body consisting of springs (the edges of the Gabriel graph), masses (the cells, vertices of the graph), and pressurized chambers (the polygons formed by the edges). We employed the Bullet library (2011), but since it was originally created to simulate rigid-body objects, forces affecting the soft-bodied animats were calculated by custom GReaNs code while the Bullet library was only used to integrate the motion of cell centers.

All cells have the same mass, and all edges have the same elasticity and damping coefficients (Hook’s coefficients). Actuation is achieved by varying the resting lengths of the springs in the structural graph. Each cell-vertex can contract or expand the elastic edges that are connected to it, provoking the shrinkage or dilation of the regions around that cell. A cell can control this process using two outputs of its GRN: one output for the contraction of the resting lengths, the other output for their expansion. Together, two cells connected by an edge modify the resting length L of that edge

additively:

$$L = (1 + A_{max} \cdot (e_1 + e_2 - c_1 - c_2)) \cdot L_0 \quad (2)$$

where e_1, e_2 (respectively, c_1, c_2) are the concentration levels of the expansion (respectively, contraction) factors in the two cells, and A_{max} is a parameter of the system representing the maximum actuation amplitude (set to 0.2 here).

Additionally, a mechanism of *pressurized chambers* is introduced in the body to oppose excessive compression and prevent collisions of internal nodes with springs. These chambers play the role of a “hydrostatic skeleton” for the animat. At the time of the transformation to the final structure, the area of each chamber is computed and defined as its equilibrium area. Then, as a chamber shrinks or expands during movement, pressure forces react along the normal of each one of its edges:

$$F_p = c_p \cdot L \cdot \left(1 - \frac{S}{S_0}\right) \quad (3)$$

where F_p is the pressure force acting outward along the normal of the edge that is considered, L is the length of this edge, S and S_0 represent the current and equilibrium areas of the chamber, and c_p is a global pressure coefficient controlling the resistance to compression.

To simulate the fluid-like environment, we apply the simplified model of fluid drag described by Sfakiotakis and Tsakiris (2006) and previously used in a work about developing spring-mass animats by Schramm et al. (2011). This model assumes that the fluid is stationary and that the force acting on a single edge of the skin is a sum of tangential and normal drag components, v_T and v_N , with respect to the motion of this edge:

$$F_T = -d_T \cdot L \cdot \text{sign}(v_T) \cdot (v_T)^2 \quad (4)$$

$$F_N = -d_N \cdot L \cdot \text{sign}(v_N) \cdot (v_N)^2 \quad (5)$$

where d_T and d_N are the fluid drag coefficients (here, $d_N = 200d_T$). Since animats are soft-bodied, the lengths of the springs change dynamically and the direction of motion of a given edge is defined as the direction of its center.

Genetic algorithm and fitness evaluation

We use here essentially the same genetic algorithm as in our previous work (Joachimczak and Wróbel, 2012), with constant population size (300), elitism, tournament selection and multipoint crossover for sexual reproduction (concerning 20% of the individuals at each generation). In GReaNs, genetic operators act at the level of the genomic elements (affecting element types, sign bits, and coordinates) and multiple elements (duplications, deletions, and crossover).

To assess the fitness, the genome is first transformed into a GRN. If the GRN does not contain a directed path (sequence of connected nodes) from at least one input element to the

output elements corresponding to cell division and animat actuation, the individual is assigned a zero fitness (it would be motionless). The development is allowed to proceed for 400 simulation steps. Cell division is terminated when the size of the embryo reaches 32 cells. Individuals containing less than three cells and individuals whose development process includes a cell division in the last 100 simulation steps of their development are assigned a zero fitness. The purpose of the latter criterion is to allow time for the morphology to equilibrate after the last cell division.

After the transformation into a soft-bodied animat, the multicellular body is immersed in the simulated physical world and allowed to equilibrate for 200 simulation steps while the GRN is stopped. This equilibration step is necessary because the levels of expansion and contraction factors in each cell at the end of development can be non-zero. Then, the GRN is started again and the animat is allowed to move for 6000 simulation steps, at the end of which the distance traveled by its center of mass is converted into a fitness value. Since absolute distance is rewarded, it is beneficial for individuals to be bigger. Indeed, we observe that the best evolved animats almost always have the maximum possible cell size and number (32 cells). The rules of physics in the environment used for development and assessment of mobility are different, but the cells can still communicate through diffusive factors during motion. This diffusion process takes into account distances between cells at the end of development.

The initial population is generated randomly, by creating positive-fitness individuals with 10 regulatory units, each unit containing one P and one G element. Most random genomes created in this fashion have a zero fitness, so it is necessary to generate a few hundred of them before a positive-fitness individual can be placed in the initial population.

Results and Analysis

We have simulated evolution in several independent runs under various environmental conditions (the physics parameters for the simulation of motion, see below). We avoided settings in which the mass of the cells was so high that it could result in exaggerated stretch to the body, or in which spring constants were so high that they would lead to instability or “unnatural” motions. Unnatural motions exploit unwanted artifacts, such as collisions of internal nodes with each other or interpenetration of body fragments (the latter could always be reduced by decreasing the time step). Under these constraints, we were still able to obtain effective patterns of locomotion over two orders of magnitude of the fluid drag coefficient d_N , and across a range of Hook’s elastic coefficients and hydrostatic skeleton pressure values c_p .

Evolution was successful at finding animats capable of locomotion. In nearly all runs, using a variety of parameters for the local physics, our genetic algorithm produced GRNs

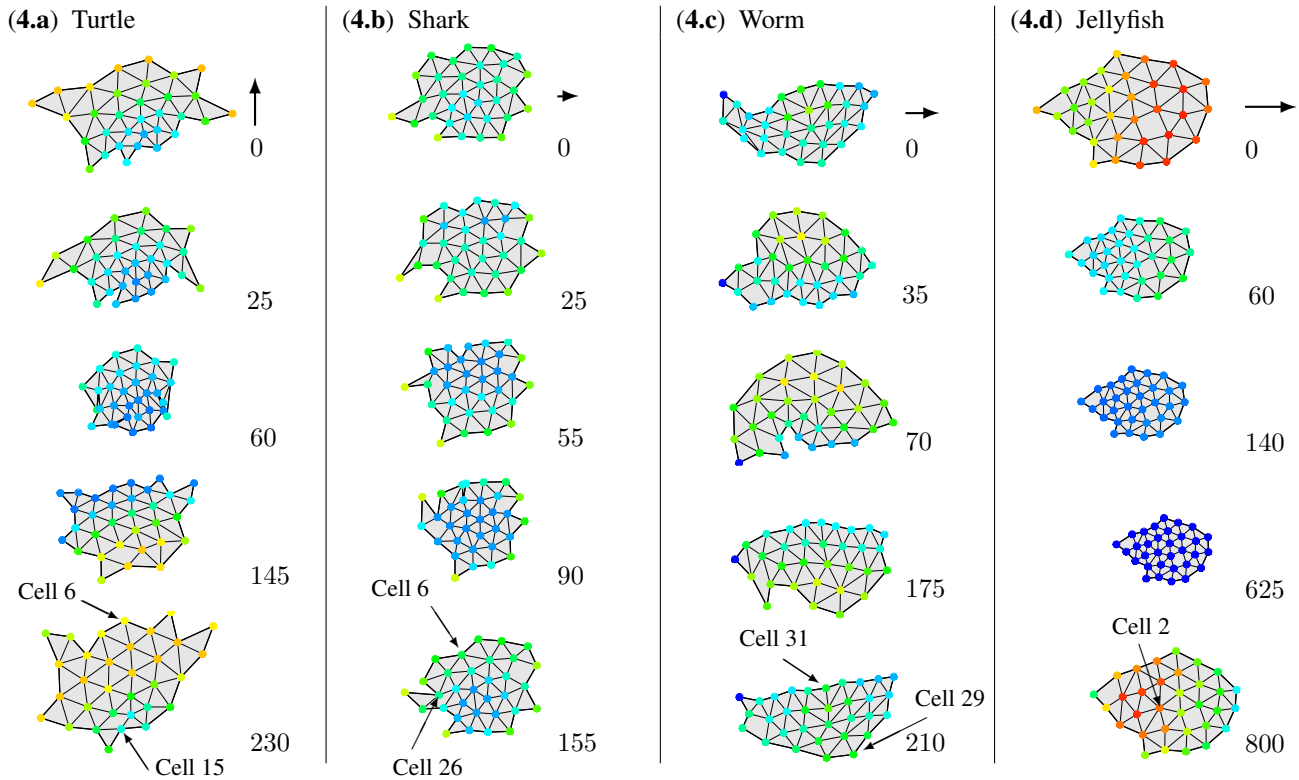
that could control both a developing animat morphology and its functional motion via coordinated contractions and expansions. In some evolutionary runs, structures that looked like “appendages” have emerged. Motion was caused by emergent oscillations and other periodic patterns controlled by the GRN in each individual cell of the animat. The results obtained here are consistent with our previous experiments in which motion was not dynamically controlled by the GRN in real time, but rather the equilibrium length of the springs and the phase and frequency of oscillations were determined and fixed at the end of development (Joachimczak and Wróbel, 2012).

To analyze the behavior of the animats, we describe them over two axes: the main body axis (front-back) and the left-right axis. These were determined by computing the direction of motion of the animat, and declaring the resulting vector (extending from the center of mass of the animat) as the main body axis, then the orthogonal direction as the left-right axis. The *activity* of each cell was defined as the absolute change in contraction or expansion of the resting length from the previous time step ($|\Delta(e_i - c_i)|$ from equation 2). The average activity along an axis was computed by projecting all cells onto this axis, and calculating the mean over the area before and after the center of mass. We will thus discuss the average cell activity of the front of the animat compared to the back, and the left compared to the right. We also show the concentrations of the expansion and contraction factors in a few selected cells of the animats, to explain how overall animat motion is generated by the collective behavior of several GRNs.

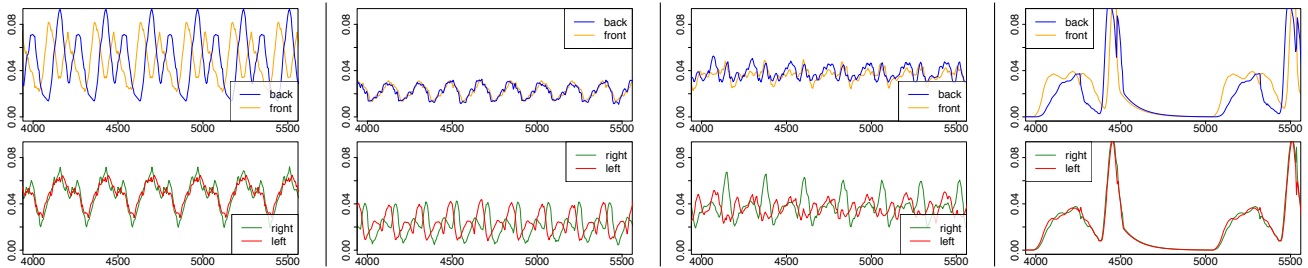
We identified several distinct strategies through which locomotion was achieved. We informally describe four such strategies here, calling them *turtle*, *shark*, *worm*, and *jellyfish*.¹ Naturally, these metaphors only refer to the visual appearance of motion, not the actual mechanism by which these real-world, nerve-endowed animals operate. Indeed, the difficulty of finding nerve-free organisms for such metaphors highlights the fact that the biological organisms that we are familiar with control their motion using nervous systems. The *worms* and *turtles* are similar to individuals seen in our previous work (Joachimczak and Wróbel, 2012). The *jellyfish* strategy, however, is new in our present control model, and the *shark* is either new or, perhaps, an extreme version of a worm-like behavior.

The *turtle* strategy is based on the use of approximately symmetric protrusions on the left and right of the animat, which move in more or less regular oscillatory patterns. Average cell activity oscillate symmetrically over the left-right axis, with changes in phase and amplitude over the front-back axis. Similar individuals constituted the majority of the best individuals obtained in independent runs under low fluid drag. In most of these individuals, the motion stemmed

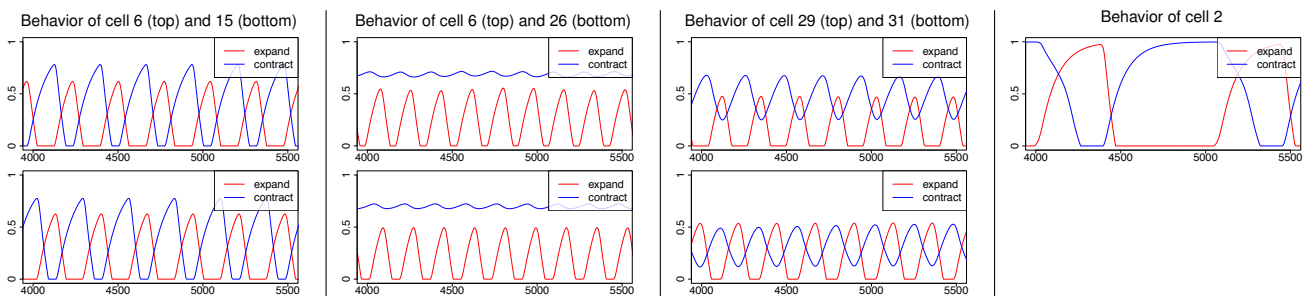
¹Supplementary videos of animat behaviors are available at: <http://evosys.org/grnanimats>



(4.1) A snapshot of motion cycles of the individuals. Node color indicates whether the cell is contracting or expanding its springs (red: expansion, blue: contraction, green: neutral). Numbers indicate the time steps in the cycle. The arrow is an approximation of the distance traveled by the center of mass in one cycle for each animat.



(4.2) Plots of the average activity of cells (*y-axis*) over time (*x-axis*) along the front-back and left-right axes of the animats.



(4.3) Plots of pattern of actuation (*y-axis*) for one or two particular cells over time (*x-axis*), where the red line indicates the concentration of the expansion factor and the blue line corresponds to the contraction factor.

Figure 4: Visualization of exemplars of the four strategies of behavior discovered by evolution.

from a wave of expansions and contractions continuously traveling from the back towards the front of the animat. The analysis of one such individual (Fig. 4.a) revealed that cells shared the same overall evolved pattern of activity. The concentration of the factors that caused expansion and contraction remained antisynchronized inside each individual cell, while there was another phase shift (almost at antiphase) when comparing the same product between different cells in the front and the back of the animat (Fig. 4.3a). Thus contractions in the front practically corresponded to expansions in the back, and vice-versa (Fig. 4.2a, top), in a manner consistent with a traveling wave of contraction-expansion across the body.

In the *shark* strategy, there was a protrusion at the back of the animat, which oscillated at a relatively high frequency with a larger displacement than the remainder of the body. The average cell activity over the front-back axis oscillated symmetrically, while there was a change in phase over the left-right axis (Fig. 4.2b). Multiple individuals of this type have been observed, even though they clearly did not exhibit an aerodynamic shape. For the individual shown in Fig. 4b, the motion was driven by a wave of expansion that traveled in the direction perpendicular to the motion, from the left to the right. However, a cell located at the tip of the motion-generating protrusion was excluded from this wave pattern and maintained a constant maximum concentration of the expansion factor, thereby sustaining the length of protrusion. Furthermore, a bulge located on the left, next to the back protrusion, collided during its own expansion with the “tail” in every cycle, passing on its kinetic energy and making the tail quickly reverse its direction of motion. Interestingly, the concentration of the contraction factor remained constant (although not uniform) in all cells, so it only provided a bias for the resting lengths of the springs. The analysis of two particular cells located at the back of the individual (Fig. 4.3b) revealed sinusoidal oscillations of the expansion factor. They had the highest oscillation frequency among all individuals investigated in this paper.

The *worm* strategy involved an elongated body driven by the propagation of synchronized waves of contraction and expansion, which traveled in the direction perpendicular to the motion, from the left side of the body to the right, resulting in undulatory movement. Cell activity here was not symmetric, neither over the front-back nor over the left-right axis, and the average activity was less regular than in other strategies (Fig. 4.2c). Only a few such individuals were observed. Comparing the activity of the expansion and contraction factors in cells located symmetrically on the left and right sides of the body (Fig. 4.3c) revealed sinusoidal oscillations in antiphase and shifted approximately by half a period between the sides of the body.

Finally, animats using the fourth strategy, *jellyfish*, were bilaterally symmetric with one blunt end and one pointed end. The whole body expanded or contracted at the same

time. Because fluid drag generated by an edge was proportional to the square of its velocity, slower expansion resulted in a smaller drag. Animats with a pointy front contracted slowly and expanded very rapidly, while animats with a pointy back expanded slowly and then contracted rapidly. In the individual of the latter type analyzed in detail (Fig. 4.2d), the compacted state was sustained and the body moved by inertia for some time, slowed down by the fluid drag, and then the cycle repeated itself. The overall impression was that of a propelling motion similar to a jellyfish. The observed pattern of cell activity resulted from the fact that the expansion factor’s concentration decreased much faster than it increased (Fig. 4.3d), and from a matching dynamics of the contraction factor. The levels of both factors in the cell were stable when the body traveled by inertia.

Throughout, we noted that evolution found synchronized actuators for contraction and expansion to great effect. However, it seemed to avoid using the full amplitude of actuation possible. Rather, it explored a trade-off between amplitude and frequency: increasing the rate of activity buildup required more products binding at high levels to a given regulatory unit.

Summary

In this work, we have re-approached the development and control of virtual soft-bodied robots in GReaNs. In contrast to our previous study (Joachimczak and Wróbel, 2012) and other models (Schramm et al., 2011), the simulations described here relied on gene regulation for both the developmental process and behavioral control. Evolution was successful at generating moving animats and discovering several functional locomotion strategies. Motion was controlled via coordinated cell actions, where individual cells displayed emergent periodic patterns of expansion and contraction. Moreover, a previously unseen form of behavior, one characterized by rapid contraction or expansion of a largely symmetric animat, was discovered. This behavior was made possible by the GRN’s fine-grained control over the contraction and expansion speeds, instead of a sine-driven actuation as in our previous work.

The reliance of the evolved locomotion mechanisms upon oscillatory changes in product concentrations is reminiscent of the rhythmic motor patterns of biological animals. By contrast, the movement of our animats is not based on a central pattern generator but a distributed collective effect. All cells of these soft-bodied, brainless animats can be potentially involved in actuation and control. It was demonstrated previously that a GRN could easily evolve toward an oscillatory behavior (e.g., Banzhaf, 2003; Joachimczak and Wróbel, 2010b). Our results show that, while motion relies on periodic changes of product concentration, development results in the differentiation of cells along the body axes in terms of phase and amplitude of these oscillations. In other terms, high evolvability stems from the relative ease of

evolving oscillatory GRNs, while a natural outcome of the developmental process is that neighboring cells have similar, though not identical dynamic properties.

The animat model used in this paper, a collection of springs modifying their resting length, is similar to a model of a soft-bodied robot. We expect that altering the physical part of the model to accommodate other types of actuation should yield similar results. In particular, the present system could be adjusted to generate designs for realistic soft-bodied robots. One of the possible directions for future work is to incorporate a notion of “energy efficiency” into the fitness function by assuming the use of a given type of existing hardware actuators.

Another direction for future work is to allow *active guidance* without a nervous system. This could be achieved for example by allowing surface cells to sense chemical gradients and modify their pattern of activity accordingly, as well as to pass information to internal cells through the use of diffusing morphogens.

One of the features of artificial life is the liberty to make counterfactual assumptions. Amongst other things, we view this work as a challenge to like-minded practitioners: qualitatively describe the role of neural machinery, and from there, refine our understanding of the role of a neural system.

Acknowledgments

This work was supported by a PAN-CNRS collaborative project, IO PAN (task IV.3), and a scholarship from the French government to BW. TK’s and RD’s positions at the ISC-PIF were funded by Région Ile-de-France. Computational resources were provided by the Polish Ministry of Science and Education (project N519 384236, N303 291234), the Tri-City Academic Computer Centre (TASK), and the Interdisciplinary Centre for Molecular and Mathematical Modeling (ICM, University of Warsaw; project G33-8).

References

- Banzhaf, W. (2003). On the dynamics of an artificial regulatory network advances in artificial life. In *ECAL 2003: 7th European Conference on Artificial Life*, pages 217–227.
- Beer, R. (1996). Toward the evolution of dynamical neural networks for minimally cognitive behavior. In *From animals to animats 4: International Conference on Simulation of Adaptive Behavior*, pages 421–429.
- Bonner, J. T. (2008). *The Social Amoebae: The Biology of Cellular Slime Molds*. Princeton University Press.
- Bullet Physics library 2.79 (2011). <http://www.bulletphysics.org>.
- Dale, K. and Husbands, P. (2009). The evolution of reaction-diffusion controllers for minimally cognitive agents. *Artificial Life*, 16(1):1–19.
- Doursat, R. (2008). Programmable architectures that are complex and self-organized: From morphogenesis to engineering. In *Artificial Life XI: Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 181–188.
- Gabriel, K. R. and Sokal, R. R. (1969). A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18(3):259–278.
- Hogeweg, P. (2000). Evolving mechanisms of morphogenesis: on the interplay between differential adhesion and cell differentiation. *Journal of Theoretical Biology*, 203(4):317–333.
- Joachimczak, M. and Wróbel, B. (2010a). Evolving gene regulatory networks for real time control of foraging behaviours. In *Artificial Life XII: Twelfth International Conference on the Simulation and Synthesis of Living Systems*, pages 348–355.
- Joachimczak, M. and Wróbel, B. (2010b). Processing signals with evolving artificial gene regulatory networks. In *Artificial Life XII: Twelfth International Conference on the Simulation and Synthesis of Living Systems*, pages 203–210.
- Joachimczak, M. and Wróbel, B. (2011). Evolution of the morphology and patterning of artificial embryos: scaling the tri-colour problem to the third dimension. In *ECAL 2009: 10th European Conference on Artificial Life*, pages 33–41.
- Joachimczak, M. and Wróbel, B. (2012). Co-evolution of morphology and control of soft-bodied multicellular animats. In *GECCO ’12: Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*. ACM. (in print).
- Kessin, R. (2001). *Dictyostelium: Evolution, Cell Biology, and the Development of Multicellularity*. Cambridge University Press.
- Lefèvre, J. and Mangin, J.-F. (2010). A reaction-diffusion model of human brain development. *PLoS Computational Biology*, 6(4):e1000749+.
- Mjolsness, E., Sharp, D. H., and Reintz, J. (1991). A connectionist model of development. *Journal of Theoretical Biology*, 152(4):429–453.
- Salazar-Ciudad, I. and Jernvall, J. (2002). A gene network model accounting for development and evolution of mammalian teeth. *Proceedings of the National Academy of Sciences*, 99(12):8116–8120.
- Schramm, L., Jin, Y., and Sendhoff, B. (2011). Emerged coupling of motor control and morphological development in evolution of multi-cellular animats. In *ECAL 2009: 10th European Conference on Artificial Life*, pages 18–26.
- Schramm, L. and Sendhoff, B. (2011). An animat’s cell doctrine. In *ECAL 2011: 11th European Conference on the Synthesis and Simulation of Living Systems*, pages 739–746.
- Sfakiotakis, M. and Tsakiris, D. P. (2006). SIMUUN : A simulation environment for undulatory locomotion. *International Journal of Modelling and Simulation*, 26:350–358.
- Stanley, K. O. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.
- Yamada, H., Nakagaki, T., Baker, R. E., and Maini, P. K. (2007). Dispersion relation in oscillatory reaction-diffusion systems with self-consistent flow in true slime mold. *Journal of Mathematical Biology*, 54(6):745–760.

The Minimal Complexity of Adapting Agents Increases with Fitness

Nikhil J Joshi¹, Giulio Tononi² and Christof Koch^{1,3}

¹ California Institute of Technology, Pasadena

² University of Wisconsin at Madison

³ Allen Institute for Brain Science, Seattle
nikhil.joshi@caltech.edu

Abstract

What is the relationship between the complexity and the fitness of evolved organisms, whether natural or artificial? It has been asserted, primarily based on empirical evidence, that the complexity of plants and animals increases as their fitness within a particular environment increases via evolution by natural selection (Bonner, 1988; McShea, 1996; Adami et al., 2000). We here derive an analytical relationship between these two quantities within an information-theoretical framework, showing that under certain conditions, complexity is a monotonically increasing function of fitness. We also simulate the adaptation of brains of digital organisms living in mazes and whose connectome evolves over 10,000s of generations in a stationary environment. We compute their circuit complexity, using an entropy-based measure (Balduzzi and Tononi, 2008). We find that their minimal complexity increases with their fitness, in line with our analytical derivation.

Introduction

It is often assumed (Bonner, 1988; McShea, 1996; Adami et al., 2000) that while evolving organisms grow in fitness, they develop functionally useful forms, and hence necessarily exhibit increasing complexity (McShea, 1991). Some, however, argue against this notion (McCoy, 1977; Hinegardner and Engelberg, 1983), pointing to examples of decreases in complexity, while others assert that any apparent growth of complexity with fitness is an admixture of chance and necessity (Carroll, 2001). One reason behind this absence of a consensus is the lack of formal or analytical definitions for complexity and fitness. While many context-dependent definitions of complexity exist (Shannon, 1949; Kolmogorov, 1965; Bialek et al., 2001; Adami et al., 2000; Tononi et al., 1994), fitness has been less frequently formalized into an information-theoretic framework (Orr, 2000). A recent computer model of simple animats evolving in a static environment (Edlund et al., 2011) found that the complexity of their brain was highly correlated with their fitness. However, no formal relation between these two quantities was derived.

The functional or structural complexity of a finite system usually has an upper bound related to the entropy of the sys-

tem. This provides a convenient tool for developing these notions without losing generality in approach.

Theory

We treat the agent as an out-of-equilibrium (Helmholtz) system (Dayan et al., 1995) trying to decipher hidden causes in the environment and adapting to them by approaching an equilibrium. Let \mathbf{x} be the sensory input configuration in a static environment and \mathbf{y} the actuator action. Treating this as a channel connecting the sensors of the agent to its output, the *mutual information* between \mathbf{x} and \mathbf{y} is

$$\begin{aligned} I(\mathbf{x} : \mathbf{y}) &= \sum p(x, y) \cdot \log \left(\frac{p(x, y)}{p(x) \cdot p(y)} \right) \\ &= \sum p(x) \cdot p(y | x) \cdot \log \left(\frac{p(y | x)}{p(y)} \right) \quad (1) \end{aligned}$$

The sensory input x presented to the agent at any time will be a function of both the current state of its local environment and of the previous actions taken by the agent. The transition probability model $p(y | x)$ can be written as an *effective* generative model $q_{\text{eff}}(x_0)$, which determines the future course of the agent, once the initial input state is specified. Thus,

$$I_q = \sum p(x_0) \cdot q_{\text{eff}}(x_0) \log \frac{q_{\text{eff}}(x_0)}{r_{\text{eff}}(x_0)} \quad (2)$$

where $r_{\text{eff}}(x_0) = p_{\text{eff}}(y)$. We drop the unambiguous indices henceforth and simply write q for $q_{\text{eff}}(x_0)$, etc. We assume that at the time-scale of evolution, performance of the agent is independent of the initial condition x_0 . We furthermore assume that the statistical properties of the environment remain constant over this time scale, that is, that the contribution due to environmental variability in q remain constant. I_q effectively quantifies to what extent the agent will try out different responses, given the same input pattern. This quantity provides an upper entropic bound on the functional complexity of the agent (Touchette and Lloyd, 2000, 2004).

The fitness of an agent is proportional to the variability in its response to a given sensory input pattern, since by sampling from its behavioral repertoire, the agent can increase

its probability of success. Thus, fitness is proportional to the size of the available repertoire-space. Entropy provides the best measure for available state-space size. In far-from-equilibrium systems, phase-space volume contraction rate is related to the entropy production rate (Crooks, 1999; Daems and Nicolis, 1999; Falkovich and Fouxon, 2004). The fundamental theorem of natural selection in evolutionary theory (Fisher, 1930; Orr, 2009), on the other hand, equates the genetic variability - the rate of the evolutionary phase-space volume change - to the fitness of a species. Indeed, evolutionary entropy offers a better measure for fitness than any other form of variance, even generalizing to dynamically changing environments (Demetrius, 1977; Demetrius and Ziehe, 2007). We here argue that fitness of our adapting agent should be proportional to the entropy of its action, or $f_q \propto -\sum q \log q$.

However, unlike complexity, which is a characteristic of the system alone, fitness is always relative to a particular ecological niche. The optimal target strategy, however, being implicit or hidden in the structure of the environment, is not directly available to the agent. Rather, the agent must optimize its *guessed* strategy, based on its own performance. If the environment can be described by a target generative model, q_T , specifying the best survival strategy, the fitness of the agent must depend on how far the *guessed* generative model q lies from the ideal or target situation-action repertoire, q_T . This is typically measured by the relative entropy $D_{KL}(q||q_T)$.¹ Hence, we assume $f_q \propto 1/D_{KL}(q||q_T)$.

Combining the two expressions, expanding $D_{KL}(q||q_T)$ and introducing a proportionality constant k yields

$$f_q = k \left(\frac{-\sum q \log q}{\sum q \log q - \sum q \log q_T} \right) \quad (3)$$

The entropy in the numerator fixes the extent of fitness-changing jumps, while the relative entropy in the denominator fixes the direction in which the jumps are the most beneficial.

Fitness as defined by equation (3) diverges near the region of optimality, that is, when $q \rightarrow q_T$. This however, does not pose any serious problem, since it is not this absolute value of the fitness but only the relative fitness between two systems which is relevant in evolution (Orr, 2009).

The measure of complexity I_q in equation (2) can be rewritten in terms of fitness f_q as

$$I_q = \frac{k'(q)}{1 + k/f_q} \quad (4)$$

¹For a given input state \mathbf{x} , possible guessed output distribution is given by $q(\mathbf{y} | \mathbf{x})$, while the correct distribution would be $q_T(\mathbf{y} | \mathbf{x})$, and $D_{KL}(q(\mathbf{y} | \mathbf{x})||q_T(\mathbf{y} | \mathbf{x}))$ would be the measure of distinguishability of a *guessed* \mathbf{y} state from a correct one. On average $D_{KL}(q||q_T)$ turns out to be a measure of average prediction error (Gossner and Tomala, 2008), giving the resolving power of the brain for correctly distinguishing two decisions (Vedral et al., 1997; Vedral, 2002), given an input state.

where $k'(q) = \sum q \log q_T$. Thus, complexity is a monotonic saturating function of fitness.

Methods

To test this hypothesis, we exploited the *in-silico* evolution experiments pioneered by Edlund, Adami and others (Edlund et al., 2011). Here simple agents evolve a suitable Markov decision process (Puterman, 1994; Monahan, 1982) in order to survive in a locally observable environment. Agents must navigate and pass through a planar maze (Fig. 1a), along the shortest possible path connecting the entrance on the left with the exit on the right. At every maze door, the agent is instructed about the relative lateral position of the next door with respect to the current position via passing an information-bit (red arrows in Fig. 1a), which is available only while the agent is standing in the doorway.

Fig. 1b shows the brain of the agents, comprising three retinal collision sensors, two lateral collision sensors, two movement actuators, and four internal reserve units for developing logic, including memory. The next-door information is received via a door-sensor. The connectome is completely specified by its genome and is kept fixed throughout the lifetime of each agent.

The evolutionary setup, based purely on stochastic mutation and driven by natural selection, allows us to monitor trends in the complexity of the brain of the agents. We use the state-averaged version of *integrated information* or Φ (Balduzzi and Tononi, 2008) of a network of interacting variables (or nodes) as a measure of complexity and relate it to the degree to which these agents adapt to their environment. The disconnected or insignificant part of the network is eliminated and the corresponding value Φ_{MC} for the largest connected part of the network or *main complex* is used for further analysis. For more details, see the supplementary text, S1.

Results and Discussion

Our numerical experiments replicated those of (Edlund et al., 2011). There is a clear trend for integrated information of the *main complex*, Φ_{MC} to grow with fitness f_q , computed relative to a perfectly adapted agent (with $f_q=100\%$). The Spearman's rank correlation coefficient is 0.75 for this dataset. The correlation coefficients for each of 126 evolutionary histories are broadly distributed (Fig. 1c). We also hand-devised an "Einstein" agent (within the constraints of the stochastic Markov networks we use), plotted as a magenta asterisk in Fig. 2.

Note that complexity and fitness were neither explicitly connected by construction nor measured in terms of each other. While the complexity of the agent's brain is completely determined by the transition table associated with its nodes, its fitness can only be evaluated by monitoring the performance of the agent in a particular environment.

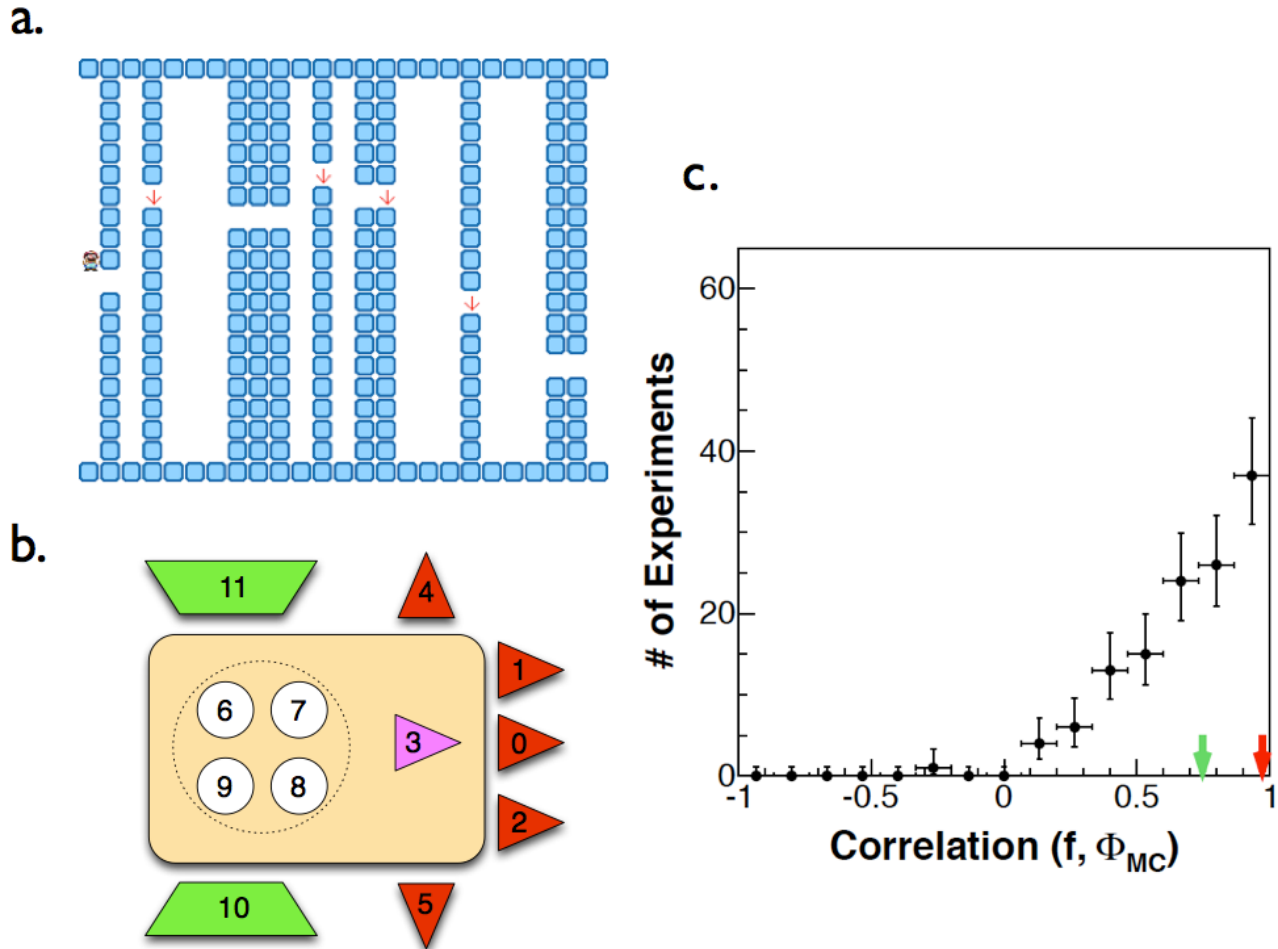


Figure 1: Experimental setup to evolve a population of agents from (Edlund et al., 2011). **a.** A section of the planar maze that the animats have to cross from left to right as quickly as possible. The arrows in each doorway represent a door bit that is set to 1 whenever the next door is on the right-hand-side of the current one. **b.** The agent, with 12 binary units that make up its brain: b0-b2 (retinal collision sensors), b3 (door-information sensor), b4-b5 (lateral collision sensors), b6-b9 (internal logic), and b10-b11 (movement actuators). In the first generation of each evolutionary history, the connectivity matrix is initiated to be random. The networks for all subsequent generations are selected for their fitness. **c.** Distribution of Spearman's rank correlation values between integrated information of the network shown in panel b, Φ_{MC} , and fitness, f_q , calculated for each of 126 evolutionary histories of 60,000 generations. The mean correlation is 0.69 with a variance of 0.24. Within each history, Φ_{MC} and fitness were evaluated after every 1000th generation. The arrows represent values reported previously (Edlund et al., 2011) (0.94; in red) and obtained for the entire, concatenated data set of Fig. 2 (0.75; in green). The distribution shows a tendency towards high correlation, yet with a large variance, indicative of additional factors not controlled for.

Closer inspection of the plot of Φ_{MC} versus fitness (Fig. 2) reveals a prominent lower boundary on Φ_{MC} for any fitness level f_q . The complete absence of any data points below this boundary, combined with the high density of points just above the boundary, implies that developing some minimal level of integrated information is *necessary* to attain a particular level of fitness. The boundary can easily be fit by our analytically derived equation (4) between entropic complexity and fitness, with two degrees of freedom. The

existence of such a boundary had been previously surmised in empirical studies (Bonner, 1988; McShea, 1996), where complexity was measured crudely in terms of organismal size, number of cell-types, and fractal dimensions in shells.

Conversely, no restriction on an upper value for Φ_{MC} is apparent in Fig. 2 (apart from the maximum level of Φ_{MC} bounded by the entropy of the animat which is 12 bits). That is, once the bound on minimal integrated information has been achieved, organisms can develop additional complexity

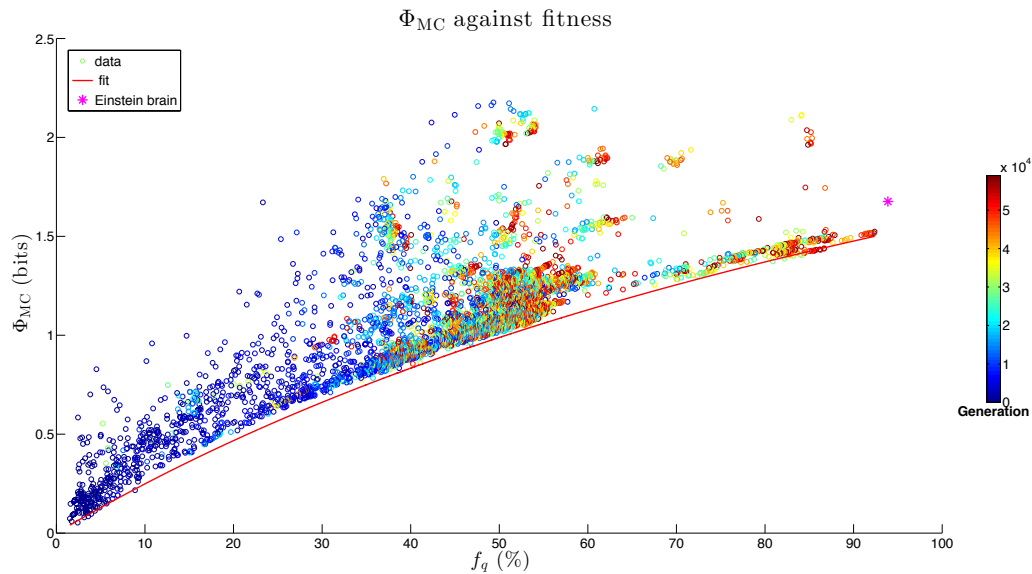


Figure 2: Data from the *in-silico* evolution study, plotting *Integrated Information*, Φ_{MC} , (in bits) against fitness, f_q for 7,560 agents sampled across 60,000 generations in 126 independent runs of evolutionary history. The Spearman's rank correlation coefficient for this data-set is 0.75 (green arrow in Fig. 1c). Fitness is computed by estimating how quickly the animats move through the mazes. The data points are color mapped according to the generation they correspond to along their evolutionary line. The curve is a numerical fit to the lower boundary using equation (4) with $k = 1.4$ and $k' = 3.8$. The *starred* point at $f_q = 93\%$ corresponds to Φ_{MC} of an optimally designed, rather than evolved, network that still retains some stochasticity.

without altering their fitness. This is an instance of degeneracy, which is ubiquitous in biology, and which might even drive further increases in complexity (Tononi et al., 1999).

We find a similar lower boundary when using predictive information, I_{pred} , rather than Φ_{MC} (see the Appendix). This supports the notion of a general trend between fitness and minimal required complexity.

Thus, complexity can be understood as arising out of chance and necessity (Carroll, 2001). The previously reported correlation between integrated information and fitness (Edlund et al., 2011) should be understood in this light. High correlation values correspond to data points close to the lower boundary. This strong correlation deteriorates as more and more data lies away from the boundary, yielding the broad distribution of the correlation values for the 126 separate histories (Fig. 1c). The additional complexity is not directly relevant for survival (though they may become so at a later stage in evolution). As a consequence of equation (4), to achieve a higher fitness, the brain of the agent must be modified either by altering its interconnections or by introducing more functional units. Conversely, to achieve a certain fitness level, a minimal level of complexity is required.

Acknowledgement: We would like to thank Jeffrey Edlund and Nicolas Chaumont for developing the evolutionary framework, Virgil Griffith for stimulating discussions and Samruddhi Ghaisas-Joshi for help with the English. This

work was funded in part by the Paul G. Allen Family Foundation and by the G. Harold and Leila Y. Mathers Foundation.

References

- Adami, C., Ofria, C., and Collier, T. (2000). Evolution of biological complexity. *Proceedings of the National Academy of Sciences*, 97(9):4463.
- Balduzzi, D. and Tononi, G. (2008). Integrated information in discrete dynamical systems: motivation and theoretical framework. *PLoS computational biology*, 4(6):e1000091.
- Bialek, W., Nemenman, I., and Tishby, N. (2001). Predictability, complexity, and learning. *Neural Computation*, 13(11):2409–2463.
- Bonner, J. (1988). *The evolution of complexity by means of natural selection*. Princeton Univ Pr.
- Carroll, S. (2001). Chance and necessity: the evolution of morphological complexity and diversity. *Nature*, 409(6823):1102–1109.
- Crooks, G. (1999). Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences. *Physical Review E*, 60(3):2721.

- Daems, D. and Nicolis, G. (1999). Entropy production and phase space volume contraction. *Physical Review E*, 59(4):4000.
- Dayan, P., Hinton, G., Neal, R., and Zemel, R. (1995). The helmholtz machine. *Neural computation*, 7(5):889–904.
- Demetrius, L. (1977). Adaptedness and fitness. *American Naturalist*, pages 1163–1168.
- Demetrius, L. and Ziehe, M. (2007). Darwinian fitness. *Theoretical population biology*, 72(3):323–345.
- Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.
- Durbin, R. (1998). *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge Univ Pr.
- Edlund, J., Chaumont, N., Hintze, A., Koch, C., Tononi, G., and Adami, C. (2011). Integrated information increases with fitness in the evolution of animats. *PLoS Computational Biology*, 7(10):e1002236.
- Falkovich, G. and Fouxon, A. (2004). Entropy production and extraction in dynamical systems and turbulence. *New Journal of Physics*, 6:50.
- Fisher, R. (1930). *The genetical theory of natural selection*. Clarendon Press.
- Gossner, O. and Tomala, T. (2008). Entropy bounds on bayesian learning. *Journal of mathematical economics*, 44(1):24–32.
- Hinegardner, R. and Engelberg, J. (1983). Biological complexity. *Journal of Theoretical Biology*, 104(1):7–20.
- Kolmogorov, A. (1965). Three approaches to the quantitative definition of information. *Problems of information transmission*, 1(1):1–7.
- McCoy, J. (1977). Complexity in organic evolution. *Journal of theoretical biology*, 68(3):457.
- McShea, D. (1991). Complexity and evolution: what everybody knows. *Biology and Philosophy*, 6(3):303–324.
- McShea, D. (1996). Metazoan complexity and evolution: Is there a trend? *Evolution*, pages 477–492.
- Monahan, G. (1982). A survey of partially observable markov decision processes: Theory, models, and algorithms. *Management Science*, pages 1–16.
- Orr, H. (2000). Adaptation and the cost of complexity. *Evolution*, 54(1):13–20.
- Orr, H. (2009). Fitness and its role in evolutionary genetics. *Nature Reviews Genetics*, 10(8):531–539.
- Puterman, M. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, Inc.
- Shannon, C. (1949). Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21.
- Tononi, G., Sporns, O., and Edelman, G. (1994). A measure for brain complexity: relating functional segregation and integration in the nervous system. *Proceedings of the National Academy of Sciences*, 91(11):5033.
- Tononi, G., Sporns, O., and Edelman, G. (1999). Measures of degeneracy and redundancy in biological networks. *Proceedings of the National Academy of Sciences*, 96(6):3257.
- Touchette, H. and Lloyd, S. (2000). Information-theoretic limits of control. *Physical review letters*, 84(6):1156–1159.
- Touchette, H. and Lloyd, S. (2004). Information-theoretic approach to the study of control systems. *Physica A: Statistical Mechanics and its Applications*, 331(1):140–172.
- Vedral, V. (2002). The role of relative entropy in quantum information theory. *Reviews of Modern Physics*, 74(1):197.
- Vedral, V., Plenio, M., Jacobs, K., and Knight, P. (1997). Statistical inference, distinguishability of quantum states, and quantum entanglement. *Arxiv preprint quant-ph/9703025*.

Appendix A. Experimental Setup

The maze is a two-dimensional labyrinth that needs to be transversed from left to right (Fig. 1a) and that is obstructed with numerous orthogonal walls with only one opening or door bored at random. At each point in time, an agent can remain stationary, move forward or move lateral, searching for the open door in each wall in order to pass through. Inside each doorway, a single bit is set that contains information about the relative lateral position of the *next* door (for e.g. arrows in Fig. 1a represent a value of 1, implying that the next door is to the right, *i.e.*, downward, from the current door). This *door bit* can only be read by the agent inside the doorway. Thus, the organism must learn that this bit represents this information that would enable it to efficiently move through the maze and it must evolve circuitry to store this information in a 1-bit memory.

The maze has circular boundary conditions. Thus, if the agent passes through the exit door before its life ends after 300 time steps, it reappears on the left side of the same maze.

Fig. 1b shows the anatomy of the agent's brain. It comprises a three pixel retina, two wall-collision sensors, two actuators, a brain with four internal binary units, and a door-bit sensor. The agent can sense a wall in front with its retina - one pixel in front of it and one each on the left and the right front sides - and a wall on the lateral sides via two collision sensors - one on each side. The two actuator bits decide the direction of motion of the agent: step forward, step laterally right- or left-ward, or stay put. The four binary units, accessible only internally, can be used to develop logical functions, including memory. The door bit can only be set inside a doorway.

While the wall sensors receive information about the current local environment faced by the agent at each time-step, the information received from the door bit only has relevance for its future behavior. During evolution of the brain of these animats, they have to learn the importance of this bit, store it internally and use it to seek passage through the next wall as quickly as possible.

The connectome of the agent, encoded in a set of stochastic transition tables or hidden Markov modeling units (Durbin, 1998; Edlund et al., 2011), is completely determined by its genome. That is, there is no learning at the individual level.

Each evolutionary history was initiated with a population of 300 randomly generated genomes and subsequently evolved through 60,000 generations. At the end of each generation (after 300 time steps), the top 10% of the agents ranked according to their fitness populate the next generation of 300 agents via point mutation. Our experiment comprised 126 such independent histories.

Fitness

The fitness of the agent is a decreasing function of how much it deviates from the shortest possible path between the entrance and exit of the maze, calculated using the Dijkstra search algorithm (Dijkstra, 1959). To assign fitness to each agent as it stumbles and navigates through a maze m during its lifetime (of 300 time steps), its fitness score is calculated as

$$s(m) = \sum_{t=0}^T \left(\frac{D_m - d_m(t)}{D_m} + N_{\text{loop}}(t) \right) \quad (5)$$

where D_m is the maximum of shortest path distances from all positions in m , while $d_m(t)$ is the shortest path distance to exit from the position of the agent at time-step t . N_{loop} counts how many times the agent has reached the exit in its life and reappeared on the left-extreme of the maze. The fitness of the agent is then the geometric mean of its fitness-score relative to the optimal score from 10 such repetitions.

$$f = \left(\prod_{i=0}^{10} \frac{s^i(m)}{s_{\text{opt}}(m)} \right)^{1/10} \quad (6)$$

Because our agents must evolve brain structures to identify the *rules* with which the environment (here, maze) has been designed, and not to develop the best strategy to solve one particular instance of it, the maze m was redesigned after every 100 generations.

Complexity

The complexity developed in the brain of an agent along the evolutionary line corresponding to the most highly adapted agent after 60,000 generations was measured in intervals of 1,000 generations for all 126 evolutionary histories. We chose the state-independent version of the integrated information measure Φ (Balduzzi and Tononi, 2008), for quantifying the complexity of the processing network of each organism. Φ reflects the co-existence of functional specialization and integration and is defined as,

$$\Phi = I(X_t : X_{t+1}) - \sum_{M^i \in \text{MIP}} I(M_t^i : M_{t+1}^i) \quad (7)$$

where MIP represents a specific way of partitioning the system X into parts M^i , such that only a minimal fraction of the total information flows across rather than inside the parts. The function $I(X : Y)$ is the usual mutual information function defined in equation (1). By definition, Φ of a network reduces to zero if there are disconnected parts, since this topology allows for a method of partitioning the brain into two disjoint parts across which no information flows. Indeed, it is only the connected part of the brain which can contribute to the information flow between sensors and actuators. As a result, we first determine the main complex (MC) for each agent by maximizing Φ . The corresponding value of the Φ is denoted as Φ_{MC} and is used for further study. For further details, see (Edlund et al., 2011).

Appendix B. I_{pred} against fitness f_q

The diminishing-demand relationship derived in equation (4) is expected to be a generic trend in any form of functional complexity with growing fitness. If that is true, the same behavior seen in case of the integrated information Φ must be observed if some alternative measure for complexity is used. One such alternative definition of complexity is the predictive information I_{pred} (Bialek et al., 2001), which is given by

$$I_{\text{pred}} = I(X_t : X_{t+1}) \quad (8)$$

where X_t and X_{t+1} correspond to states in which the system is observed at time t and $t + 1$ respectively. In short, I_{pred} quantifies the predictive power of an agent in terms of the dependence of its responses at time $(t + 1)$ on the input sensory pattern presented at an earlier time-step or at time t . Fig. 3 shows the I_{pred} values, when plotted against corresponding fitness f_q .

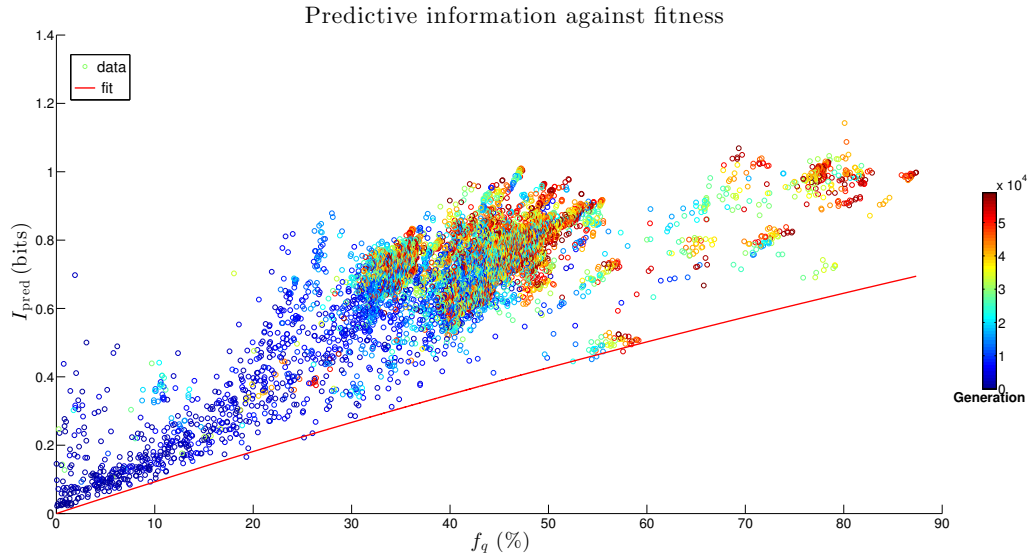


Figure 3: Results from the *in-silico* evolution study, for I_{pred} against fitness, f_q . The data points are color mapped according to the generation number they correspond to along their evolutionary line. The curve shows fit to the boundary with the relation in equation (4).

We observed a similar trend as in Φ_{MC} , though less prominent, confirming that the relationship between fitness and the minimal required complexity is a generic characteristic of evolving complexity. It must be noted that a similar trend has been demonstrated via empirical studies (Bonner, 1988; McShea, 1996), in case of organismal sizes, cell-type variety. The trends in these studies were shown mainly against evolutionary period rather than against increasing fitness.

On the Emergent Behaviors of a Robot Controlled by a Real-Time Evolving Neural Network

Walter O. Krawec¹

¹Stevens Institute of Technology, Hoboken, NJ 07030
wkrawec@stevens.edu

Abstract

In this paper we apply a real-time evolving neural network which uses a hill-climbing algorithm capable of adapting not only a network's synaptic weights but also its topology (creating a recurrent neural network). We then apply this network to a robot in a simulated environment. By equipping the robot with a minimal set of instincts and a short-term memory system (to facilitate reinforcement learning), we observe that several strategies developed which pass the emergent behavior test of (Ronald et al., 1999). In particular, we see robots learning behaviors that are not rewarded by the environment.

Of course a hill-climbing algorithm is more likely than a genetic-algorithm to get stuck at a local optimum, we argue that, despite this, the method described here has several unique advantages. In particular, it allows us to create a single persistent robot that slowly learns and "grows up" as described in (Ross et al., 2003). With our system, it is an individual that learns not a population of individuals, and our learning is continual (e.g. there is no need to reset the robot to some starting position to evaluate the fitness of a particular network).

We conclude with several future problems and applications. For instance, we describe a simple mechanism allowing a network to be copied to embedded hardware whenever a network connection is available to a PC (which is responsible for the memory and time intensive task of evolving the network). This mechanism does not require a continual link to a PC. We also discuss the possibility of creating a distributed evolving neural network system.

Introduction

In this paper, we apply a real time evolving neural network (ENN) to a (currently simulated) robot which begins its existence without any prior knowledge of itself or its environment. That is, the robot has absolutely no idea as to what its various inputs mean, or what its outputs do. By equipping the robot with a simple short term memory, along with a variety of very basic "instincts" and "reflexes", we allow the robot's neural network to evolve itself (in realtime). This evolution adjusts not only synaptic weights, but also the network's topology (creating a fully recurrent network).

To accomplish this, a hill-climbing approach is used instead of the more typical genetic algorithm (see (Junfei et al.,

2007), (Floreano & Keller, 2010), (Stanley & Miikkulainen, 2002), (Cliff et al., 1992), and (Stanley et al., 2005) for some examples). By using a hill-climbing approach, we believe we achieve more interesting personalities (term used informally of course; the point is, that by using one network and slowly adapting it, certain traits unique to a particular robot are preserved throughout its lifetime - traits that may be lost if a multi-generational GA were used); furthermore, it allows us to achieve certain items mentioned in (Ross et al., 2003) listing the conditions required for allowing a robot to "grow up". Furthermore, our algorithm does not require multiple agents to be evaluated and then reset (or even a single agent to be run, evaluated, then reset); instead a single robot may operate while continually evolving as it gains new experiences.

Evolution is facilitated by equipping each robot with a short-term memory (STM). This memory stores recent actions along with the rewards/penalties given for their performance. These rewards/penalties are provided by a robot's minimal instincts. Furthermore, STM is not a look-up table. Indeed it may be that some entries are incorrect and others are never filled. Furthermore, STM only holds data for a (relatively) short time; it is up to the ENN to remember the data and avoid situations that result in penalties (such as crashing into walls).

Our simulations will consist of one or more robots equipped with a variety of sensors (e.g. proximity, light, sound etc.). We use a very minimalist reward system as driven by simple instincts. Specifically we use three instincts: Pain, Boredom, and Human Response (allowing a human operator to train a robot to perform a specific task if so desired). By implementing these few rewards, our robot's neural network is able to quickly learn to avoid walls. However other very interesting behaviors may emerge including:

1. Closely following a wall (despite binary proximity sensors).
2. Learning to follow a second robot which has learned to avoid walls
3. Some robots learn to group together while others avoid

each other

4. Learn to seek, avoid, or ignore light

We use the emergent behavior test defined by (Ronald et al., 1999) for our purposes and show that many unexpected strategies developed by our robots pass this test. Informally, however, the behaviors developed are considered to be emergent due to the fact that such strategies are not directly rewarded (and hence not expected). For example, there is no reward for grouping together, yet many robots developed behaviors that, in addition to avoiding obstacles, sought out other robots in the simulation. We will describe this in more detail later.

We also demonstrate other advantages to our real time ENN approach. This includes the ability of allowing a robot to adjust to new I/O in real time (where the new I/O may be added “on-the-fly” while the robot is in operation). We make use of this ability to slowly train a robot to perform more complicated tasks from simpler ones (another condition of (Ross et al., 2003)). Also our ENN very easily evolves to memorize certain patterns in input data which may be beneficial to future work.

Real Time Evolving Neural Network

We now describe briefly the evolution algorithm used in our experiments. Given a neural network \mathcal{N} , we assume that there exists a fitness function f mapping \mathcal{N} to $f(\mathcal{N}) \in \mathbb{R}$ such that $f(\mathcal{N}_1) > f(\mathcal{N}_2)$ implies that \mathcal{N}_1 is a “better fit” to some data set than \mathcal{N}_2 . We formalize this later by constructing such a fitness function based on a robot’s short term memory. In this section however we simply describe the evolution of a neural network with respect to this function f .

Our algorithm begins with a simple feed-forward network consisting of a single neuron for each input and output. Furthermore, input neurons are assigned linear activation functions while output neurons (and indeed any other hidden node) is assigned a sigmoid function (specifically $\frac{1}{1+e^{-x}}$). Also, if requested, our network may begin with a collection of random hidden neurons. These neurons are connected randomly to the network.

A single iteration of the evolution algorithm will take as input a network \mathcal{N} along with fitness function f and output a network \mathcal{N}' (possibly the same network) such that $f(\mathcal{N}') \geq f(\mathcal{N})$. This is achieved by taking N copies of \mathcal{N} and modifying each (independently) according to the following rules:

1. If a neuron or synapse was added or removed within the last T evolution cycles, we modify 15% of the synaptic weights. This is to allow changes to the network’s topology to “settle” optimally.
2. With probability p_1 , we add a new random neuron

3. With probability p_1 we remove a random neuron (but not I/O neurons)
4. With probability p_2 we add a random synapse connecting two neurons (chosen at random; possibly the same neuron creating a loop)
5. With probability p_2 we remove a random synapse
6. With probability $1 - 2p_1 - 2p_2$, we modify 15% of the synaptic weights.

From these $N + 1$ networks (for we consider the original \mathcal{N}), we choose $\frac{N+1}{K}$ of the very best (those with the highest value according to f) and $\frac{N+1}{2K}$ random networks to which we apply a second iteration of the evolution algorithm to each of these separately (and indeed, this recursion repeats a total of R times). Note that if a neuron/synapse was added or removed in any iteration, further applications of the evolution algorithm will simply adjust synaptic weights. Finally, we choose the very best of the resulting $N + 1$ networks and output it. The original network \mathcal{N} is replaced by this new network.

For our experiments, we found good results by setting $p_1 = 0.004$, $p_2 = 0.006$, $N = 100$, $R = 1$, and $K = 10$. Of course larger values of N , K , and R should lead to faster learning though it does slow the algorithm.

Furthermore, every change made to a network is logged in a simple evolution tape. By following this tape, we may separately re-create the evolution of a network. This permits us to create a multi-threaded application where one thread is devoted to running the evolution algorithm while another thread keeps an independent copy for use in real-time. Every few cycles, the evolve tape may be requested (rather, only the latest changes made to the evolve tape). From this tape, the networks may be synchronized.

Additionally this evolve tape allows us to easily run an evolving neural network on embedded hardware. This is accomplished by using a PC to evolve a network (which requires substantial time and memory) while the embedded device simply requests the latest evolve tape every so often. From this tape, the embedded device may very easily build a local copy of the network to run at will. This (and other applications of the evolve tape) are described in a later section.

Short-Term Memory

In the previous section we described the evolution algorithm with respect to a fitness function f . We now describe how this function is actually constructed.

Every robot is equipped with a form of short-term memory (STM). This memory is responsible for holding a limited amount of “possibly useful” information that a neural network should attempt to capture. We say “possibly” useful since it may be that a particular action was mistakenly

added to STM. Hence our mechanism must be able to guide the evolution of a neural network but not maintain its contents for too long a time. To achieve this, we propose two methods: *U-Learning* and *EL-Learning*. We will describe their application and also mention some of the advantages and disadvantages to each.

We begin with *U-Learning* (the *U* stands for *utility*). This method is similar to *Q-Learning* (Watkins & Dayan, 1992) in that we will store a matrix mapping state/action pairs to rewards. However the method differs in that *U-Learning*'s goal is only to exist in the short-term whereas *Q-Learning*'s goal is to fully explore the reward space.

We begin with a zero matrix $U^0 \in \mathbb{R}^{2^m \times 2^n}$ where m is the number of inputs to our network and n is the number of outputs. The superscript is used to index the time at which this U matrix is valid. Given utility matrix $U^t = (u_{i,j}^t)$, the value $u_{i,j}^t$ represents the reward (or utility) for applying action j from state i . States and actions are computed in the obvious way: given input vector $(x_0, x_1, \dots, x_{m-1}) \in \{0, 1\}^m$ and output vector $(y_0, y_1, \dots, y_{n-1}) \in \{0, 1\}^n$, then the state (respectively action) is simply $\sum x_i 2^i$ (respectively $\sum y_i 2^i$).

When in operation, if a robot receives a reward or penalty $r \in \mathbb{R}$ (exactly how a robot receives these rewards/penalties is described later) for performing a certain action J given state I , then we construct a new matrix $U^{t+1} = (u_{i,j}^{t+1})$ where:

$$u_{i,j}^{t+1} = \begin{cases} u_{i,j}^t + r & \text{if } i = I \text{ and } j = J \\ u_{i,j}^t & \text{otherwise} \end{cases}$$

Finally, since we are only interested in storing data in the short-term, every T cycles, we construct the matrix: $U^{t+1} = \eta U^t$ where $\eta \in (0, 1)$. For our experiments, we set $T = 10$ and $\eta = 0.8$.

From all this, we may construct our fitness function with respect to U^t . This function, which takes as input a neural network \mathcal{N} (with m inputs and n outputs) and *U-learning* matrix U^t is defined in pseudocode as:

```

function UFitness( $\mathcal{N}$ ,  $U^t$ )
    sum = 0
    for  $j = 1$  to  $M$  do
        Choose  $\pi_j$  a random permutation of  $\{0, \dots, 2^m - 1\}$ 
        for  $i = 0$  to  $2^m - 1$  do
            Run  $\mathcal{N}$  on input representing state  $\pi_j(i)$ 
            for  $k = 0$  to  $2^n - 1$  do
                 $p = u^t(\pi_j(i), k)$   $\triangleright u^t(x, y) = u_{x,y}^t$ 
                for  $l = 0$  to  $n - 1$  do
                    if  $l$ 'th bit of  $k$  is 1 then
                         $p = p \times (l$ 'th bit of output of  $\mathcal{N}$ )
                    else
                         $p = p \times (1 - l$ 'th bit of output of  $\mathcal{N}$ )
                    end if
                end for
            end for
        end for
    end for

```

```

        sum = sum + p
    end for
end for
end for
return (sum/ $M$ )

```

Note that we must randomly permute the states else a network evolves to expect inputs in order 0, 1, etc. We average the fitness over M distinct permutation. M of course should depend on the state size of a network, for our experiments we found good results with $M = 10$.

We now present a second method of handling a robot's STM which we call *EL-Learning* (the *EL* stands for *Evolution List*). This method begins with an empty list $E^0 = \emptyset$ (again the superscript determines the time index at which this list is valid). Elements in E^t will consist of 4-tuples of the form $(\mathbf{x}, \mathbf{y}, r, T)$ where $\mathbf{x} \in \mathbb{R}^m$ (where m is the number of inputs to our ENN), $\mathbf{y} \in [0, 1]^n$ (n being the number of outputs), $r \in \mathbb{R}$ is the reward for outputting \mathbf{y} given input \mathbf{x} , and T is the lifetime of this element (measured in evolution cycles; if $T = \infty$ its lifetime is infinite).

Whenever a reward is received, we construct a new list $E^{t+1} = E^t \cup \{(\mathbf{x}, \mathbf{y}, r, T)\}$ where \mathbf{x} is the current input vector, \mathbf{y} the current output vector, r the reward value (possibly negative implying a penalty) and T is this element's lifetime (depending on the type of reward, in our experiments this value ranges from 50 – 200). Furthermore, if $|E^{t+1}| \geq M$ for some M we remove the element from E^{t+1} with the smallest value of T . This is done not only to keep our list at a manageable size, but also to keep its purpose as a short-term memory mechanism - not a look-up database. It is possible to set $M = \infty$ which implies the list will continue to grow with elements removed only if $T = 0$.

Finally the fitness function with respect to E^t is defined as:

$$f_{E^t}(\mathcal{N}) = \sum_{e \in E^t} e.r(n - \delta(\mathcal{N}(e.\mathbf{x}) - e.\mathbf{y})), \quad (1)$$

where $\delta(x, y)$ is the usual Euclidean distance squared.

After running our experiments multiple times, we saw that the resulting ENNs evolved using *U-Learning* or *EL-Learning* behaved differently. Due to the nature of the evolution process, ENN's not only evolve to achieve a higher fitness function, but they also tend to "remember" the order the STM data is sent to it. Because of this, *EL-Learning* will typically create a network that learns to expect its input in the order presented in the list and will perform different actions given a different input order (though we may undo this by choosing a random permutation as with the *U-Learning* fitness function). Since with *U-Learning*, this order is randomized (and indeed this is the reason for choosing a random permutation; else a network learns to expect its inputs in order 0, 1, . . .), the resulting network is usually more "stable". However *EL-Learning* does tend to produce more interesting behaviors.

Furthermore, EL-Learning permits us to easily use analog inputs whereas *U*-Learning is binary by nature. We tested this by developing a simple game where an ENN is in charge of shooting at an enemy ship (this enemy ship is floating in space and controlled by a human operator). The inputs to the network are the enemy ship's x and y velocities and its x and y coordinates. The output of the ENN is a value between 0 and π which is translated to the gun's rotation. After watching a human point the gun for a short time (less than a minute with less than 100 rounds), the ENN is able to very accurately point the weapon (usually after running the evolution algorithm for only 20-100 iterations). This demonstrates an ENN's ability to learn a continuous space with only minimal training.

Finally, *U* learning requires maintaining in memory a rather large matrix (though since most of the elements are zero as we show later, memory may be saved by simulating it as a list) however the evolution tends to be faster and more stable. With further work, we believe that EL-Learning should not only be able to reliably create a stable ENN but also allow for more interesting behaviors to emerge.

Reflexes, Instincts, and Decision Paths

We now answer the question as to how data is inserted into STM. Each robot is equipped with a very minimal set of reflexes and instincts. Reflexes, when triggered (either by the environment, instincts, or by the ENN itself), take full control of the robot for a short amount of time before returning control back to the ENN. A reflex may only control a robot's outputs and may be used to avoid dangerous situations (such as turning away from a wall we just crashed into), or to help the ENN with complicated motion tasks (e.g. moving a leg forward on a legged robot).

Instincts are also minimal subroutines however they may only be triggered by the environment and they do not directly control a robot (though they may trigger a reflex). When triggered, an instinct will provide a reward or penalty to the robot's STM. It is from this data that the robot uses to evolve. Examples of instincts include the before mentioned pain, boredom (to prevent a robot from performing the same action for too long), and human input (to allow a robot to be taught from a human).

Of course instincts are useful to provide rewards or penalties for certain instantaneous actions (e.g. penalizing the action of moving straight when a forward pointing proximity sensor reports an obstacle); however there might be several "decisions" made by an ENN before an instinct was triggered that led to this reward or punishment. Also, we should provide some small reward for actions that do not lead to an instinct being triggered. To this end, we introduce "decision paths" which is a secondary form of STM. Essentially, this mechanism will log a robot's actions in some time interval $[t_i, t_{i+1}]$ (where $t_0 = 0$, and $t_{i+1} = t_i + T$ for some $T > 0$). This log is a list of state/action pairs (s_i, a_i) where

a_i was the action taken by the ENN at state s_i (note that if EL-Learning is used, s_i and a_i are vectors). An element of this form is added to the list at fixed intervals $T' < T$ and potentially also whenever a state or action changes. A decision path is therefore an ordered list $P = \{(s_i, a_i)\}$ where the state action pair (s_i, a_i) occurred in time before (s_j, a_j) for all $i < j$.

When T units of time have expired, we take our decision path $P = \{(s_i, a_i)\}$ and add to primary STM (either a *U*-Learning matrix or an EL-Learning list) the triple (s_i, a_i, r_i) where $r_i = \eta^{|P|-i}$ with $\eta \in (0, 1)$. That is, since we've avoided triggering an instinct we classify the last actions as good however at a discounted factor the further in the past they occurred.

If, however, an instinct is triggered with reward value $w \in \mathbb{R}$ before T units of time have elapsed, we then take our decision path P and add to primary STM the triple (s_i, a_i, r_i) where $r_i = w\eta^{|P|-i}$ with $\eta \in (0, 1)$. That is, our discount factor now depends on the instinct's reward value.

Hence, instincts that provide a penalty will result in any action leading up to it to be considered *potentially* incorrect (the further in the past, the less this is penalized) and likewise any instinct that provides a positive reward will result in actions leading up to it to be potentially correct. Since STM slowly loses data over time, any incorrect rewards or penalties added will eventually be discarded or replaced by newer data as the robot discovers it.

Evolving in Stages; or The Persistence of Long-Term Memory

One concern the reader may have is that our ENN (which may be considered the robot's Long-Term Memory) will lose its memory once the STM does. That is, if given two fitness functions f_A, f_B where B is a proper subset of A , is it the case that $f_A(\text{evolve}(\mathcal{N}, B)) \ll f_A(\mathcal{N})$ assuming \mathcal{N} has been evolved over time using A ; i.e. does evolving a network with respect to B (which has strictly less data than A) cause our network to completely disregard prior information learned from A ?

While we are still investigating this, it doesn't seem to matter very much that the STM only retains partial reward information. Over time, as the STM loses its memory and the ENN continues to evolve over the degraded STM set, the network may "forget" certain actions. However it seems to quickly recover its memory when presented with the proper reward. Furthermore, due to the decision path system, if a robot doesn't use a particular sensor (or subset of sensors) for some time, the ENN may forget what action to take when those sensors are used again. None the less, it seems to be the case that the robot can quickly recover that knowledge. Besides, this is a problem that other living organisms face (e.g. humans).

Finally, we point out that in operation the STM only holds a small fraction of reward information yet this allows the

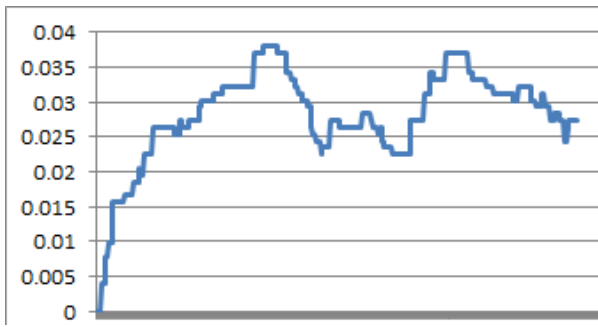


Figure 1: Graph of a robot's STM contents over time (x -axis). The y axis plots the proportion of the U -matrix entries that are non-zero. We notice that at the start of the simulation, the contents of STM quickly grows (as the robot crashes into obstacles for example), peaks, then dissipates (as the robot settles on a strategy/behavior). The memory peaks again when new information is received (an instinct being triggered for example due to an unexpected circumstance or loss of long-term memory). Note however that the content in these simulations doesn't exceed 4 percent of all possible state/action pairs. This is in line with the goal of STM - it is not to serve as a general look-up table for every possible action but meant only to guide an ENN's evolution.

robot to perform very well in an environment. See Figures 1 and 2 for a graph of the STM's contents over time (using U -Learning).

We also note that we are able to develop a robot in stages. For example, we may permit a robot to run on its own for some time learning to use its proximity sensors. Then, after this has been learned, we may place it in a new environment and/or teach it to use a different subset of sensors. Left on its own, a robot can usually quickly learn not to crash into walls however it may ignore its other sensors. However we may, at any time, "teach" it to perform certain actions with its other sensors. How exactly this teaching is accomplished is described later (we permit the human operator to use only minimal signals/hints). Also, how this affects STM is shown in Figure 2.

Advantages to using a Hill-Climbing Approach

In (Ross et al., 2003), the authors described 5 conditions allowing a robot to "grow up". They are:

1. It is the individual that develops rather than a system of individuals
2. Involves acquiring a hierarchy of skills where the acquisition of new skills is facilitated by already acquired ones.
3. Not necessary to learn one skill at a time; learning is continual

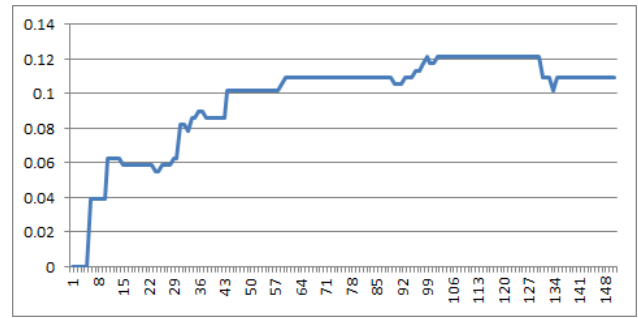


Figure 2: Similar to figure 1 except using a robot with a smaller input set. Here we note that again the STM contents quickly fill then stabilize around 11 percent. The increase around time 90 is when we attempt to teach a robot a new task. Once this task is learned, the STM once again decreases (around time 135) to 11 percent.

4. The reason why one action is preferred over another changes with experience
5. In the process of development, the individual becomes capable of purposeful action on longer time spans.

Of course there is the obvious problem that a hill-climbing (HC) approach may more likely settle at a local optimum (though many humans also do this - we call it being "stuck in a rut"). However by having a network slowly grow by only altering its structure slightly at each iteration (as opposed to a genetic algorithm (GA) where several different competing structures are considered and at any time, a network may be replaced by one that is radically different) we satisfy condition 1.

Furthermore, because a network slowly grows in this fashion, certain behaviors in a robot may develop and present themselves, then later lie dormant only to reappear much later in the robot's life. This happened in our simulations multiple times. For one particular example, we had a robot that learned to avoid walls but would, at regular intervals, perform a zig-zag motion (note that there was no reward for this particular action; it was just a developed personality unique to this particular robot). This behavior eventually disappeared as the robot continued to evolve (satisfying condition 4). However, much to our surprise, this zig-zag motion reappeared much later in the simulation. Whatever neural structure that created this behavior remained and was able to re-emerge later; something that would be unlikely in a GA where we might have thrown away this particular population member. A HC algorithm however maintains much of a robot's past. We return to the other conditions of growing up later.

There may be some interesting future work combining the genetic algorithm and hill-climbing approach. Indeed, we may begin by using a GA to evolve a decent population;

each robot may then use one of the networks. Then by using a HC approach as described above, we permit the robot to continually learn and grow.

Experiments

We tested our design in a simple simulated environment. A robot is controlled by a single ENN which communicates with the simulator over a standard network connection (hence the simulator may run on a different computer). If an experiment calls for multiple robots running within the same simulated environment, each of these robots has its own ENN and STM. In fact, each robot runs as its own process and the only communication between these robot processes is through whatever indirect means is available in the simulator (e.g. crashing into one another).

A robot consists of two motors (differential drive), two microphones (left and right; we discuss their purpose later), a bump skirt, and one of the following additional configurations:

1. Three binary IR proximity sensors (left, right, and center)
2. Three binary IR proximity sensors, two light sensors (also binary)
3. Three binary IR proximity sensors, four robot detectors (these are able to determine if another robot is nearby in a certain direction; having four of these allows the robot to detect when a robot is ahead, behind, left, or right).
4. Four robot detectors (but no proximity sensors)

Each sensor (besides the microphones and bump skirt which are only processed by the instincts not the ENN directly) has its own input into the ENN and each motor its own output (hence configuration (2) has 5 inputs and 2 outputs). Each robot is equipped with the following instincts and reflexes:

1. Repulse Reflex: Moves the robot backwards and turns randomly
2. Crash Instinct: When the robot physically touches something, a negative reward is learned and the repulse reflex is triggered
3. Boredom Instinct: When the robot has been performing the same action for too long a negative reward is learned.
4. Sound to Left (respectively Right) Instinct: When the robot “hears” a clap to its left, it will move to the left (respectively right) randomly and add a positive reward for doing so.

We stress that, at the start of the simulation, the robot has really no sense of left or right, forward or backward. The “sound to left/right” instinct is provided so as to allow a human to train a robot to perform certain tasks (or just to help

it along from time to time). It may seem strange that here we make a knowing distinction between left and right, however we justify it by observing that such instincts appear naturally in many organisms on this planet, so it is not such a stretch to use it here.

Using these configurations and instincts, we’ve discovered the following behaviors develop through multiple simulation runs:

- Configuration 1:
 - i) Basic obstacle avoidance
 - ii) Wall following
 - iii) Searching for obstacles by scanning left/right every so often
- Configuration 2:
 - i) Obstacle avoidance while seeking, avoiding, or ignoring light
- Configuration 3:
 - i) Obstacle avoidance
 - ii) “Follow-the-Leader” when at least one other robot was in the same simulation
 - iii) Group together, or avoid each other
- Configuration 4:
 - i) “Follow the leader” if another robot of configuration type 1-4 was in the simulation (hence following this other robot resulted in not crashing)

In (Ronald et al., 1999), the authors defined the emergence test as follows. Involving a system designer and an observer, the test proceeds in three stages:

1. Design: The system has been constructed by describing local elementary interactions between components in a language \mathcal{L}_1 .
2. Observation: The observer (who knows \mathcal{L}_1) describes global behaviors and properties of the running system over time using a language \mathcal{L}_2 .
3. Surprise: \mathcal{L}_1 is distinct from \mathcal{L}_2 ; furthermore the causal link between elementary interactions described in \mathcal{L}_1 and the behaviors actually observed in \mathcal{L}_2 is non-obvious to the observer (who is therefore surprised)

In our case, \mathcal{L}_1 is simply the three instincts along with the one reflex mentioned above. Interactions here are those that will maximize a robot’s reward value. The only thing that may diminish the reward is crashing into a wall. The language \mathcal{L}_2 consists of those behaviors mentioned above.

Of course basic obstacle avoidance is an expected behavior (hence doesn’t pass the emergence test). Wall following however was remarkable considering that the proximity

sensors used had no notion of distance (despite this some robots learned to follow the walls very closely; others from a greater distance). Also there is no reward for following a wall (only a reward for not touching it). Hence we claim this passes the test. We've also seen robots that learn to scan for walls by moving forward a certain amount then sweeping left then right. They used this strategy to follow walls closely. Again this behavior is emergent from the test.

We were also excited to see robots learning to follow each other or to group together in configuration 3. Again, there is no direct reward for doing so; hence this is an emergent behavior.

Configuration 4 learning to follow another robot (which is able to avoid walls) does not pass the emergent test (since following the other robot is the only expected strategy). Still it was an interesting result so we mention it here.

We note that the majority of the time, robots learn the very basic wall avoidance strategy. Also, robots seem to develop these emergent behaviors with or without human intervention (via the "clapping" instinct). Though it is interesting to note that more complicated behaviors seem (according to our simulations) more likely to develop if there is an occasional "helping hand" from a human supervisor (directing the robot away from a wall for example or when they get stuck in a corner; while the robot would eventually find its way out of such positions, the process is expedited by a few "claps").

Video recordings of some of these behaviors are available at our website: http://www.walterkrawec.org/robots/paper_alife13.html

Adapting to Change

We mention briefly that our robots seem to be very capable of adapting to change. While more work needs to be done investigating this area, we mention that a robot is able to compensate for a faulty sensor (e.g. a binary proximity sensor that is inverted). Also, when the sensor is restored, the robot is able to return to normal fairly quickly. All of this in real time while the robot is running. Of course compensating for faulty I/O is a quality shared by other neural networks, we were pleased to see that our STM architecture allowed for this compensation (instead of constantly enforcing old behaviors).

Additionally, we are able to add I/O "on the fly". This is accomplished simply by inserting extra I/O neurons (these are neurons that cannot be removed by the evolution algorithm) - after some iterations of the evolution process, assuming there is STM information for this new I/O, the neurons are incorporated into the network. We experimented with this by teaching a robot to use a collection of short-range proximity sensors then, when these have been learned, adding additional longer-range proximity sensors. The robot was able to learn to incorporate the new sensors while still maintaining the ability to use the original.

Applications of the Evolution Tape

We mention briefly some of the applications of the evolution tape. As already mentioned, it allows us to create a multi-threaded version of the ENN program thereby permitting one thread to work constantly on the evolution algorithm while another simply runs the produced ENN in real-time. We then use the evolution tape to synchronize the two threads' networks.

Secondly, it allows an ENN to run on embedded hardware with limited memory. The embedded device will, on occasion, request the latest evolve tape (or rather the updated section since its last request) from a PC. The PC is responsible for the actual evolution of the network. Furthermore, the embedded device will send to a PC whatever reward information it receives (via its instincts). Note that this is different from having a simple wireless network to the robot from which a PC both runs and evolves an ENN. By permitting an ENN to run locally on the robot itself, it may leave the range of the network, while also storing whatever reward information it receives (instincts should run locally on the embedded hardware). Then when back in range, the robot may send its reward information to the PC (which will incorporate it into its STM for fitness evaluations) and also request the latest evolve tape. This system does not require continual wireless communication.

We demonstrated this ability using a Parallax Propellor¹. This is an 8 core (though our system currently uses only one core) MCU with 32KB of RAM and, on our prototype board, a 5MHz clock rate. Though much work needs to be done with this; it proves that an ENN may be run (relatively easily) on embedded hardware.

We are also very interested in designing a distributed evolution algorithm. In such a setup we will allow multiple computers to each have a copy of some network \mathcal{N} and independently run a single iteration of the evolution algorithm (each computer shares a copy of the STM). When finished, the network with the highest fitness value will be chosen as the output. Whichever computer has this network will send its evolve tape to the others (again, only the portion listing the modifications made to the network which is at most $20(R + 1)$ bytes, where R is the previously defined recursion level used) allowing each to easily be synchronized. Such a setup will permit us to explore a larger section of the solution space. Furthermore, we may then use the before mentioned technique to allow the network to be quickly transferred to embedded hardware and run in real time.

Summary and Future Work

In this paper, we experimented with controlling a robot using a real-time evolving neural network and observed that many of the behaviors that presented themselves passed the

¹Propellor is a registered trademark of Parallax, of Rocklin, CA

emergence test of (Ronald et al., 1999). We argued that using a hill-climbing approach as opposed to the more typical genetic algorithm allows a robot to grow continually as an individual. That is, instead of replacing a robot's controller whenever a new population member is born, a robot slowly grows from itself. This is important for satisfying (Ross et al., 2003)'s checklist of conditions for allowing a robot to "grow up". Furthermore, a robot's past behavioral history (and personality "quirks") tends to be preserved through the evolution process.

We've already mentioned that our system satisfies conditions (1) and (4) of this checklist. Condition (3) is also satisfied since it is clear that a robot's learning is continual and we may teach a robot one skill at a time or even add new I/O to our robot who will then learn to use it. Condition (2) (that a robot learn a hierarchy of skills with new skills facilitated by older ones) we also believe is within our reach however more experimentation is required. Condition (5), which requires that the individual becomes capable of purposeful action on longer time spans, we think is also possible with our system however we have not yet constructed experiments that last long enough (we ended the majority of our simulations after 30 minutes of wall-clock time). This remains a future problem to investigate.

We also think that the EL-Learning system can be improved to take further advantage of the ENN's ability to easily memorize patterns in the received input. We believe the current mechanism doesn't promote this to its full potential at the moment.

Other work includes improving the efficiency of our learning algorithm and also to devise a distributed learning system. We also intend to take advantage of the ability to easily transfer an ENN onto embedded hardware to design a physical robot.

Finally we would like to experiment with the GA/HC hybrid approaches we mentioned before.

Acknowledgements

The author would like to thank the anonymous reviewers for their suggestions.

References

- Cliff, D., Harvey, I., Husbands, P. (1992) Incremental evolution of neural network architectures for adaptive behaviour. *Technical report CSRP256. University of Sussex School of Cognitive and Computing Sciences.*
- Floreano, D., Keller, L. (2010) Evolution of Adaptive Behaviour in Robots by Means of Darwinian Selection. *PLoS Biol.* 8(1): e1000292. doi:10.1371/journal.pbio.1000292
- Junfei, Q., Zhanjun, H., Xiaogang, R. (2007) Q-Learning based on neural network in learning action selection of mobile robot In *Automation and Logistics, 2007 IEEE International Conference*, pages 263–267. Jinan, China
- Stanley, K. O., Miikkulainen, R. (2002) Efficient Reinforcement Learning through evolving network topologies In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002)*, 9, San Francisco.
- Stanley, K. O., Bryant, B. D., Miikkulainen, R. (2005) Real-time neuroevolution in the NERO video game *IEEE Transactions on Evolutionary Computation* 9(6), pages 652–668.
- Ross, P., Hart, E., Lawson, A., Webb, A., Prem, E., Poelz, P., Morgavi, G. (2003). Requirements for getting a robot to grow up *Advances in Artificial Life Vol 7. 7th European Conference, ECAL 2003, Dortmund, Germany.* pages 847–856.
- Ronald, E., Sipper, M., Capcarrère, M. (1999) Testing for emergence in artificial life In Floreano, D., Nicoud, J., Mondada, F., editors, *ECAL '99 Proceedings of the 5th European Conference on Advances in Artificial Life*, pages 13–20. Springer-Verlag, London.
- Watkins C., Dayan, P. (1992) Q-Learning *Machine Learning*, Vol. 8. pages. 279–292.

Automatically Designing and Printing 3-D Objects with EvoFab 0.2

Tim Kuehn¹ and John Rieffel¹

¹Union College, NY 12308
rieffelj@union.edu

Abstract

Although Evolutionary Design has had great success in creating virtual objects, very few of these evolved designs have been manufactured. Standing in the way is the *fabrication gap* caused by a reliance on prescriptive rather than descriptive representations of evolved objects. Evolutionary Fabrication describes an alternative process which evolves *how* rather than *what* to build. In this paper we describe EvoFab 0.2, a completely automated physically embodied machine which implements Evolutionary Fabrication and evolves three dimensional objects. We describe the mechanism and underlying algorithms in detail, and show how it can be used to create novel structures.

Introduction

Evolutionary algorithms have been used to design a wide variety of objects, from furniture (Funes and Pollack, 1998; Hornby and Pollack, 2001) to architectures (Hemberg and O'Reilly, 2004) to robots (Sims, 1994). Evolved designs are often characterized by the novelty of their solutions, enabled by a process which operates orthogonally to human design methodologies and biases. Koza has justly described genetic algorithms as “automated invention machines” capable of human-competitive patentable designs (2003).

A historically valuable aspect of the patent process is the “working model”, a physical prototype of the design submitted to the patent office. And yet most evolved objects are never physically manufactured, relegated instead to the virtual drawing board. Those few exceptions which have been manufactured – most notably Lohn *et al.*’s antennae(2005), and Pollack *et al.*’s robots (2001) – were done so with considerable human effort and interaction (Funes’ LEGO structures, for instance, often had to be assembled sideways on a flat surface before being tilted into position.) The goal of our research into Evolutionary Fabrication is to automate the entire process of design and manufacture, leading to the possibility of a real “automated invention machine”.

One significant source of the gap between evolved design and manufactured object – what we call the *Fabrication Gap* – is the fact that almost all evolved designs are *descriptive*

rather than *prescriptive*. That is, much like a blueprint they specify what to build, but leave out the essential information of *how* to build it. Bridging the gap between evolved design and manufacture in a *post-hoc* manner requires considerable human input, and runs the risk of re-injecting human bias into a process whose success is otherwise greatly increased by the absence of such bias. Furthermore, purely descriptive evolutionary design runs the risk of generating *unbuildable* designs (Rieffel, 2006).

A second obstacle to the physical manifestation of evolved designs is the infamous Reality Gap (Jakobi et al., 1995). Evolutionary algorithms are experts at exploiting their substrates. When the design of objects happens in simulation, successful candidates often achieve high fitness by exploiting bugs in the simulator (see for instance Sim’s seminal work on evolved artificial agents (1994)). Moreover, many complex systems, including the deposition of viscous materials at the heart of rapid prototyping, cannot be simulated with any degree of transferable verisimilitude even by advanced techniques such as computational fluid dynamics.

The solution we propose lies in evolving *how* to build rather than *what* to build. In Evolutionary Fabrication, the evolving genotype explicitly describes a *process* of manufacture rather than an *object* of design. Furthermore, guided by Rodney Brooks’s sage advice that “the world is its own best model” (1990), we eschew simulation entirely, and evolve objects exclusively in the real world. In this sense we are in the company of others who have performed the evolution exclusively in the real world (Watson et al., 1999; Thompson, 1996; Zykov et al., 2004).

In 2010 we introduced EvoFab 0.1, a machine which implemented many aspects of Evolutionary Fabrication, but featured an interactive GA, and therefore required substantial human subjective interaction (Rieffel and Sayles, 2010). In this paper we describe a significant leap in the state of the art with EvoFab 0.2 (pictured in Figure 1), the first machine capable of closed-loop, fully automated Evolutionary Fabrication. EvoFab operates by embedding a genetic algorithm directly within an off-the-shelf rapid prototyping machine. The genotypes of the system are linear strings of printer in-

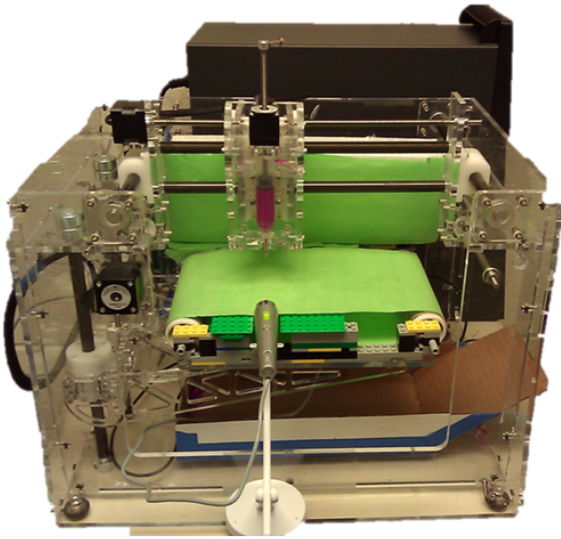


Figure 1: The “EvoFab 0.2” consists of a Fab@Home printer, computer vision software to determine fitness, and a conveyor belt, all controlled by an evolutionary algorithm.

struction primitives and phenotypes are the objects which result from printing. Fitness is determined by using machine vision algorithms to measuring physical properties of the printed objects. The process is further automated with the addition of a conveyor belt to discard objects once they have been evaluated.

EvoFab 0.2

As illustrated by Figure 2, EvoFab 0.2 is a three-stage process. First an object is printed by extruding material onto a platform. Next, the object is then evaluated using machine vision techniques. Finally, the object is moved off the printing platform to begin the process anew.

The Fab@Home Printer

At the heart of EvoFab 0.2 lies Fab@Home, an open-source 3D printer (Malone and Lipson, 2007) and appealing because of its low cost and relative ease of use. The most current model of the Fab@Home is Model 2 (Lipton et al., 2009), however we used the Model 1 as the basis for EvoFab because it allows greater access to the underlying API.

The Fab@Home operates by extruding material through a syringe and onto a platform. The carriage that holds the syringe is free to move along the X- and Y-axes. The platform upon which the material is deposited is free to move along the Z-axis. Fab@Home normally builds its products by interfacing via USB to a program that contains STL-based blueprints of objects. However, it also allows for direct control of print functions via serial port.

Printer Commands as Genotypes

Conventionally, when operating as a pure 3-D printer, Fab@Home constructs objects via additive manufacturing: depositing material layer-by-layer in a rastering process. We place no such constraint upon the operation of the printer, however. Instead, evolved genotypes are purely *prescriptive*, consisting only of a linear sequence of primitive instructions sent to the printer.

The specific instructions available as components of the genome are as follows:

- **extrude** – This command causes a small amount of material to be deposited onto the print platform.
- **beginExtrude** – This command, rather than send a command directly to the printer, controls the action of the other commands. When activated, all other commands except endExtrude will send their command coupled with an extrude command. Effectively, all other commands say “do this while extruding” when beginExtrude is activated.
- **endExtrude** – This command deactivates beginExtrude.
- **goUp** – Raises the print platform.
- **goDown** – Lowers the print platform.
- **goLeft** – Causes the print carriage to move left.
- **goRight** – Causes the print carriage to move right.
- **goIn** – Causes the print carriage to move toward the back of the Fab@Home.
- **goOut** – Causes the print carriage to move away from the back of the Fab@Home.

Print Media

While the Fab@Home is able to extrude plastic from long spools, allowing for long print durations without refilling material, plastic printing involves incredibly high temperatures, necessitating caution and constant vigilance by the user. Because this negates the ability of EvoFab to act autonomously, we chose instead to use other materials.

EvoFab 0.1 used silicone bath caulk as a print material. With new goals, however, come new requirements, and after attempts with plastic and silicone caulk, we settled on a brand of modeling compound similar to Play-Doh. Silicone caulk is easily extrudable, readily available, and comes in many colors, which is useful in allowing computer vision software to easily differentiate a printed object from its background. However, it is also sticky when first printed, and its cure time of approximately thirty minutes for faster-drying variants is too long to wait between prints. Thus, the material would inevitably stick to the print platform, making automation difficult.

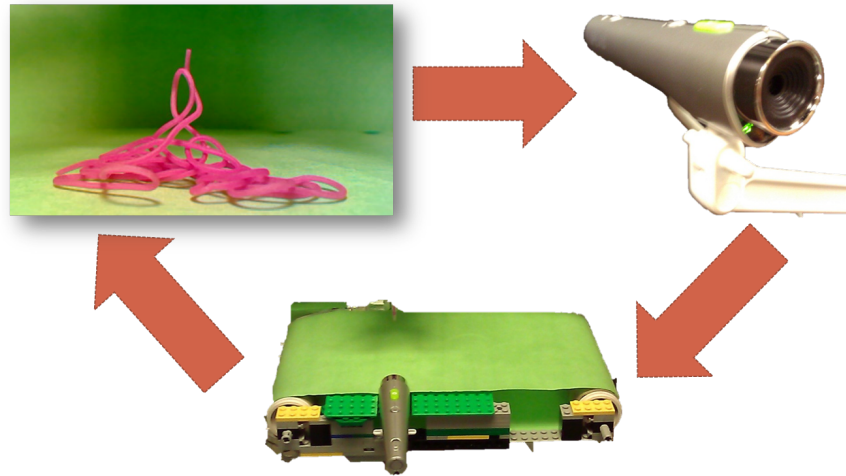


Figure 2: A graphical representation of the three-stage process: print, evaluate, recycle.

Play-Doh has the same benefit of being readily available in many colors and easily extrudable without the drawback of stickiness upon first being extruded. This lack of stickiness comes with its own set of problems: when printing, if the material is not extruded quickly enough, it will not stick to the platform, causing the print carriage to drag the thread of material around instead. This has led to a certain degree of unpredictability, but it has proven to be the best option that has been tried thus far.

From 0.1 to 0.2: Full Automation

Our earlier system, EvoFab 0.1, was the first to instantiate Evolutionary Fabrication, but suffered from several drawbacks, most notably its reliance upon subjective human input for fitness evaluation, and its reliance on human effort to clear the build platform between generations. As a result of this human involvement, a single generation of a GA could take several hours.

EvoFab 0.1 ran an interactive GA, or blind-watchmaker algorithm. The process began with the fabber printing four objects onto a piece of wax paper lying on top of the print platform. Then, a human operator would inspect the four objects and choose the one they deemed to be best-fit. Because of the complexity of the evaluation task, fitness criteria were relatively simple, such as the object's similarity to a desired 2-D letter shape ("O" or "A"). The user would then remove the wax paper containing the objects, input their fitness into a computer, and begin the cycle anew with the printing of four more objects (videos of this process are on the authors website.) While workable, there are a variety of ways in which this method was restrictive.

Automating Evaluation In an interactive GA such as the one used for EvoFab 0.1, a person's opinion on which object

is best-fit is both highly subjective and prone to error. Especially with early generations, it may be very difficult for a person to choose between four seemingly shapeless masses.

To address this in EvoFab 0.2, we developed a completely automated evaluation process. Using openCV wrapped in Python, we have created computer vision software that works in tandem with a camera affixed to the front of the printing platform. This allows for the unbiased and consistent evaluation of fitness, and further allows us to evaluate more three-dimensional objects.

Cycling Another issue that arose in EvoFab 0.1 is that comparing more than four genotypes becomes unwieldy. The print platform was originally divided into quadrants, allowing one object to be printed in each quadrant. Because the printer was set up to only print four objects at a time, it would require replacing the print platform wax paper after every four prints.

To automate this process in EvoFab 0.2, we developed a belt based upon simple lego motors driven by a USB interface. Once an object has been evaluated, it is moved off of the platform by the conveyor and deposited into a disposal container. Thanks to this improvement, EvoFab can now run unattended for approximately thirty minutes before requiring a refilled syringe, and excluding these refills, EvoFab can in principle run unattended indefinitely.

Evolving Arches

With these pieces in place, as a proof of concept we can demonstrate how the new EvoFab 0.2 can be used to evolve objects in a closed-loop manner.

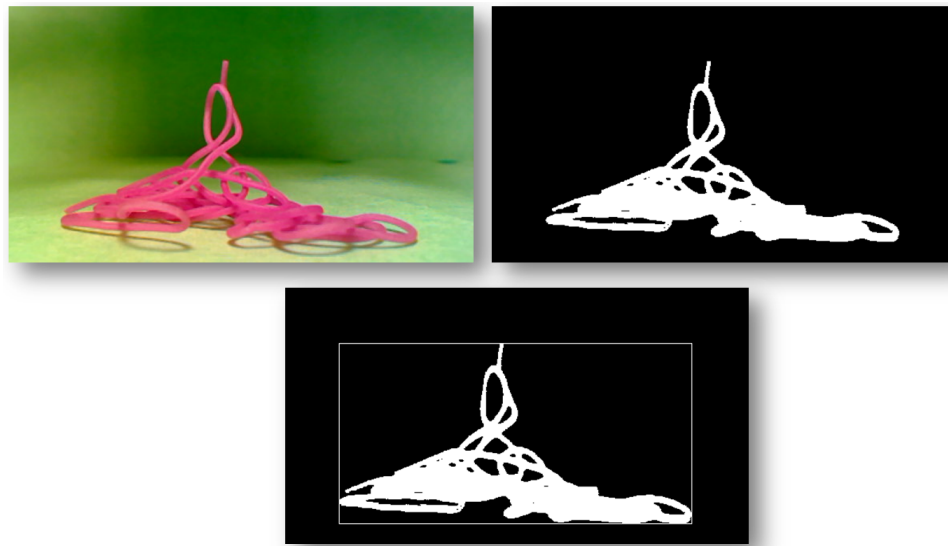


Figure 3: Evaluating "archiness": Fitness is determined by machine vision. First the image is thresholded into black and white. Second, a bounding box is calculated around the object. Finally, the percentage of overhanging mass is determined by columnwise counting the number of black pixels "shaded" beneath white pixels, and normalizing by bounding box area.

Fitness Criterion

In order to provide EvoFab with a challenge, we chose a fitness criterion which is deliberately difficult for 3-D printers to produce: overhangs. Since rapid prototypers conventionally print objects by rastering upwards layer by layer, higher layers require support from lower layers, and only very modest overhangs are allowed. As a consequence they cannot construct an objects with a large degree of cantilevering. Consider, for example, an arch, whose supporting columns are relatively easy to produce, but whose middle section can be a challenge, since it cannot deposit material onto mid-air. Our interest is therefore in discovering how a evolutionary algorithm, faced with this task, might arrive at a solution.

For our purposes, the "archiness" of a printed object can be calculated by the degree of overhang present. Figure 3 shows how such a fitness is evaluated: an image is captured by a camera that views the printing stage. Then, the image is thresholded so that the printed object is white and the background is black. This is made simple by printing in a color negative to that of the background, in this case pink being the negative of green. Then, a bounding box is drawn around the contours of the white image. For all pixels contained within the bounding box, fitness increases for every black pixel that is vertically below a white pixel in its column. Fitness is then normalized by total pixels within the bounding box to account for different sized objects, returning the percentage of overhanging mass in the image.

Evolvability of Linear Encodings

The printing of 3-D objects by extruding material from a print-head is an explicitly linear and serialized process. This

has significant consequences in terms of the evolvability of any encoding of any such process.

In a conventional GA, mutation can occur anywhere in the genome with equal probability, and at least in principle the effects of a mutation are largely independent of where along the the genome a mutation occurs. This is no longer the case in linear encodings such as ours: the effect of a mutation is highly sensitive to where in the process the mutation occurs.

Consider a simple set of instructions to draw the letter "L" in our printer language:

```
beginExtrude
goOut
goOut
goOut
goRight
goRight
goRight
endExtrude
```

A change early on in the sequence, for instance changing the first goOut to a goUp, would result in the entire shape being printed on a higher plane, which, depending upon what was underneath might drastically affect the shape, whereas changing the last goRight to a goUp would have very little effect. This dependence on context is compounded when you take into account the full three-dimensional nature of the objects being printed.

This context-sensitivity is even more pronounced when considering the effects of crossover. The building block hypothesis (Goldberg, 1989) holds that crossover aids evolution by finding and duplicating useful regions of the genome.

In a serialized encoding such as ours, however, the context dependency means that a sequence of instructions which is highly fit in one context is unlikely to be as fit in a different context. For instance, the sequence of three `goRight` commands in the example above may draw a nice straight line on a flat surface, but would have a significantly varied phenotypic consequence if executed in mid-air or over pre-existing structure.

We will elaborate on future alternatives to linear encodings in our discussion section below.

Algorithmic Details

Given the challenges to evolvability imposed by the nature of our serial encoding, we have chosen to eliminate crossover and implement a 1+4 Random Mutation Hill-climber (RHMC) (Mitchell, 1996) rather than a more canonical GA.

Initial Genome length was 350 instructions, with a 10% mutation rate. Mutation was capable of changing the operation at a locus (i.e. from `goLeft` to `goRight`), inserting a new random instruction, or deleting the current instruction.

Results and Discussion

A video of the EvoFab 0.2 in action can be found on YouTube, tagged with “`evofab`” and “`alife13`”.

Figures 4 and 5 show typical results of evolution achieved after ten generations. Raw images are in the left hand column, thresholded and bounded images are shown in the middle column, and fitness values are shown in the right hand column.

Quantitatively we can show that fitness according to our metric has increased over the course of evolution. The highest fitness individual from the original population, as measured by normalized overshadowed pixels, is 0.17, whereas the fitness of the best individual from generation 10 is 0.34.

Qualitatively, of course, the results are more equivocal. While the last figure may have a fitness measured at 0.34, it doesn’t exhibit many features which we could describe as truly arch-like. These results therefore, while promising, and clearly proof of the concept, highlight several improvements required of our system.

The most obvious problem lies in our vision-based fitness function. As written, parts of the structure which are not overhanging at all, for instance the long stretches of material on the right hand side of the middle images in Figures 4 and 5, are awarded fitness by exploiting a trick of perspective view. Regions of an object which are printed further back on the surface of the arena contribute more toward this ersatz fitness than those printed toward the front. This perspective issue can be resolved largely by changing the angle of the camera and the field of view of the vision algorithm, such that the fitness function can only measure parts of the structure which are truly overhanging.

Secondly, since our camera can only view the X-Y projection of objects, it would be worth removing or reducing the Z-axis degree freedom for the print head. An arch is an arch so long as it is upright – its thickness is largely irrelevant.

Ultimately, we expect future progress of EvoFab to hinge on the matter of genotypic representation. One possibility would be to use a grammatical encoding such as an L-System (1990) to indirectly “grow” a linear encoding - this could allow for increased modularity and reuse in the phenotype, although the resulting linear representation might still be susceptible to the context-sensitivity demonstrated by our current encoding. Alternatively we could use a developmental approach, such as the CPPNs used to by Clune and Lipson (2011) and Auerbach and Bongard (2010). While these particular efforts used CPPNs to create descriptive 3-D blueprints rather than prescriptive instructions, it would not be unreasonable to use a CPPN to directly control a 3-D printer.

Conclusion

We have described the world’s first completely closed loop system capable of automated design and fabrication and proven the concept by demonstrated its application to a complex design task. While our methods would benefit from some modification, we are confident that the approach shows promise.

In the near term, there are more immediate and practical applications of Evolutionary Fabrication than the invention of objects. For instance, every new material used by a 3-D printer requires careful calibration of flow rates and printer head speed in order to produce a consistent “bead” of material. Currently, this calibration process is entirely driven by human trial and error across a set of more than twenty parameters. We envision a simple GA being able to more quickly and effectively arrive at these calibration settings, and perhaps being able to find particularly efficient bead characteristics. Secondly, sharp corners are very difficult to produce on 3-D printers, and small errors on the corners of tall objects can quickly add up in a deleterious manner. EvoFab could be well used to discover and optimize new methods of producing corners which are less susceptible to these errors.

In the long term, the automated design of 3-D objects has valuable applications in fields ranging from biomedical applications (for instance the development of compliant soft grippers) to soft robotics, not to mention the much-vaunted “automated invention machine”. We look forward to making progress toward these goals with our upcoming EvoFab 0.3.

References

- Auerbach, J. E. and Bongard, J. C. (2010). Evolving cppns to grow three-dimensional physical structures. In *Proceedings of the 12th annual conference on Genetic and evolutionary compu-*

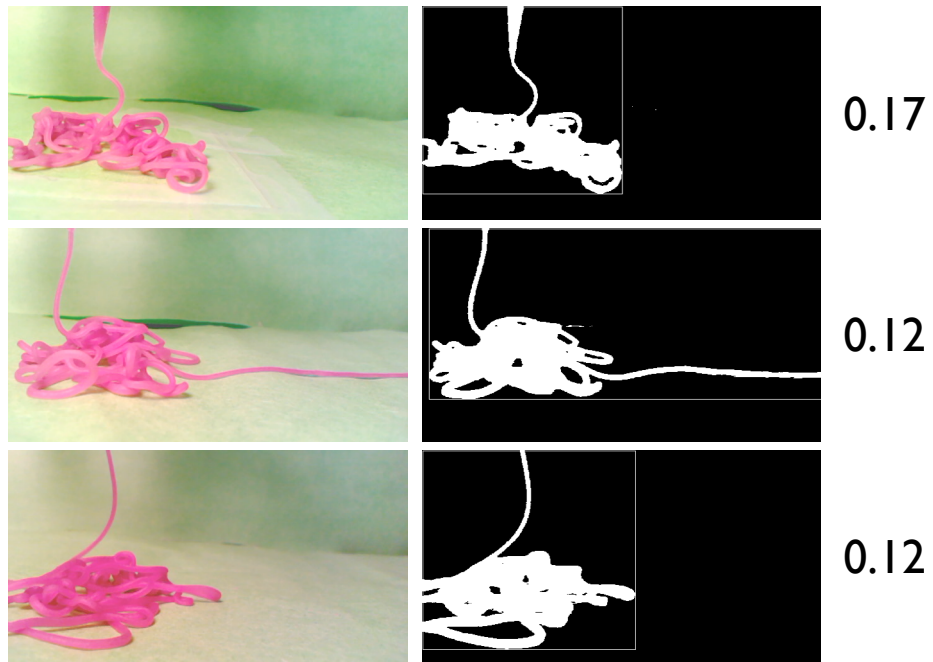


Figure 4: Example phenotypes from Generation 0. Raw images are in the left hand column, thresholded and bounded images are shown in the middle column, and fitness values are shown in the right hand column.

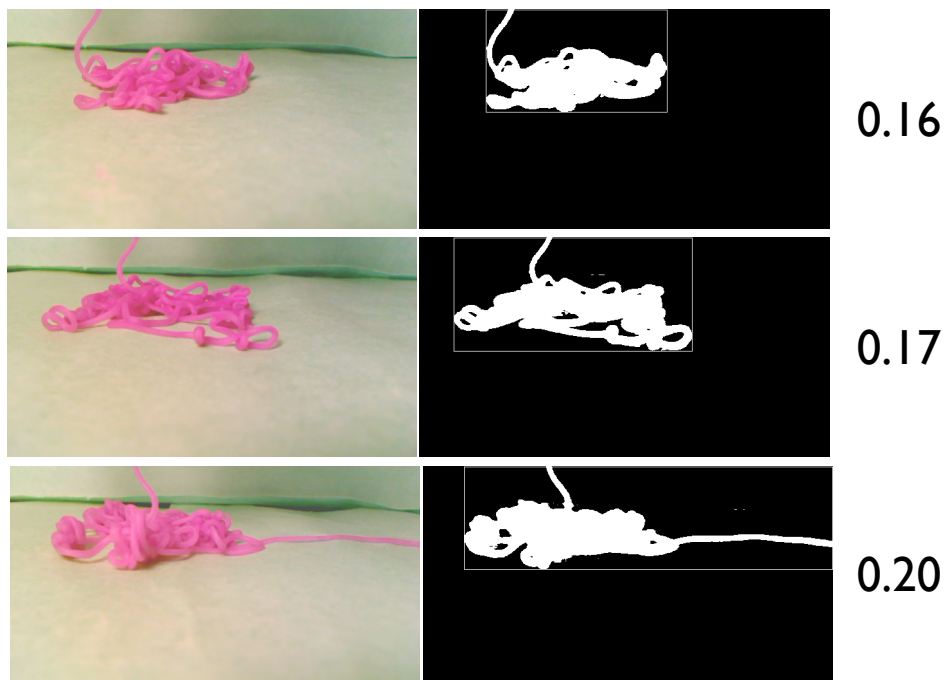


Figure 5: Example phenotypes from Generation 10.

- tation, GECCO '10, pages 627–634, New York, NY, USA. ACM.
- Brooks, R. A. (1990). Elephants don't play chess. *Robotics and Autonomous Systems*, 6:3–15.
- Clune, J. and Lipson, H. (2011). Evolving 3d objects with a generative encoding inspired by developmental biology. *SIGEVOLUTION*, 5(4):2–12.
- Funes, P. and Pollack, J. B. (1998). Evolutionary body building: Adaptive physical designs for robots. *Artificial Life*, 4(4):337–357.
- Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Professional, Upper Saddle River, NJ, USA.
- Hemberg, M. and O'Reilly, U.-M. (2004). Extending grammatical evolution to evolve digital surfaces with genr8. In *EuroGP*.
- Hornby, G. S. and Pollack, J. B. (2001). The advantages of generative grammatical encodings for physical design. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 600–607, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea. IEEE Press.
- Jakobi, N., Husbands, P., and Harvey, I. (1995). Noise and the reality gap: The use of simulation in evolutionary robotics. In *Proc. of the Third European Conference on Artificial Life (ECAL'95)*, pages 704–720, Granada, Spain.
- Koza, J. R., Keane, M. A., Streeter, M. J., Mydlowec, W., Yu, J., and Lanza, G. (2003). *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Kluwer Academic Publishers.
- Lipton, J. I., Cohen, D., Heinz, M., Lobovsky, M., Parad, W., Bernstein, G., Li, T., Quartiere, J., Washington, K., Umaru, A.-A., Masanoff, R., Granstein, J., Whitney, J., and Lipson, H. (2009). Fab@home model 2: Towards ubiquitous personal fabrication devices. In *Solid Freeform Fabrication Symposium*, pages 70–81.
- Lohn, J. D., Hornby, G. S., and Linden, D. S. (2005). An Evolved Antenna for Deployment on NASA's Space Technology 5 Mission. In O'Reilly, U.-M., Riolo, R. L., Yu, T., and Worzel, B., editors, *Genetic Programming Theory and Practice II*. Kluwer.
- Malone, E. and Lipson, H. (2007). Fab@home: The personal desktop fabricator kit. *Rapid Prototyping Journal*, 13(4):245–255.
- Mitchell, M. (1996). *An introduction to genetic algorithms*. MIT Press, Cambridge, MA, USA.
- Pollack, J. B., Lipson, H., Hornby, G., and Funes, P. (2001). Three generations of automatically designed robots. *Artificial Life*, 7(3):215–223.
- Prusinkiewicz, P. and Lindenmayer, A. (1990). *The Algorithmic Beauty of Plants*. Springer-Verlag, New York, USA.
- Rieffel, J. (2006). *Evolutionary Fabrication: the co-evolution of form and formation*. PhD thesis, Brandeis University.
- Rieffel, J. and Sayles, D. (2010). Evofab: a fully embodied evolutionary fabricator. In *Proceedings of the 9th international conference on Evolvable systems: from biology to hardware, ICES'10*, pages 372–380, Berlin, Heidelberg. Springer-Verlag.
- Sims, K. (1994). Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 15–22. ACM Press.
- Thompson, A. (1996). Silicon evolution. In *Stanford University*, pages 444–452. MIT Press.
- Watson, R. A., Ficici, S. G., and Pollack, J. B. (1999). Embodied evolution: Embodying an evolutionary algorithm in a population of robots. In Angeline, P. J., Michalewicz, Z., Schoenauer, M., Yao, X., and Zalzal, A., editors, *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 335–342, Mayflower Hotel, Washington D.C., USA. IEEE Press.
- Zykov, V., Bongard, J., and Lipson, H. (2004). Evolving dynamic gaits on a physical robot. In *Proceedings of Genetic and Evolutionary Computation Conference, Late Breaking Paper, GECCO'04*.

Rewarding Reactivity to Evolve Robust Controllers without Multiple Trials or Noise

Joel Lehman, Sebastian Risi, David B. D'Ambrosio, and Kenneth O. Stanley

University of Central Florida, Orlando, FL 32816

jlehman@eecs.ucf.edu, risi@eecs.ucf.edu, ddambro@eecs.ucf.edu, kstanley@eecs.ucf.edu

Abstract

Behaviors evolved in simulation are often not robust to variations of their original training environment. Thus often researchers must train explicitly to encourage such robustness. Traditional methods of training for robustness typically apply multiple non-deterministic evaluations with carefully modeled noisy distributions for sensors and effectors. In practice, such training is often computationally expensive and requires crafting accurate models. Taking inspiration from nature, where animals *react* appropriately to encountered stimuli, this paper introduces a measure called *reactivity*, i.e. the tendency to seek and react to changes in environmental input, that is applicable in single deterministic trials and can encourage robustness without exposure to noise. The measure is tested in four different maze navigation tasks, where training with reactivity proves more robust than training without noise, and equally or more robust than training with noise when testing with moderate noise levels. In this way, the results demonstrate the counterintuitive fact that sometimes training with no exposure to noise at all can evolve individuals significantly more robust to noise than by explicitly training with noise. The conclusion is that training for reactivity may often be a computationally more efficient means to encouraging robustness in evolved behaviors.

Introduction

A significant challenge in artificial life and evolutionary robotics (ER) is to evolve robust controllers for robots or artificial creatures (Nolfi and Floreano, 2000). While natural organisms are remarkably robust (i.e. they function over a wide range of environmental conditions), controllers evolved in simulation are often fragile and dependent upon overly specific simulation details (Jakobi, 1998; Koos et al., 2010). For example, a practical manifestation within ER of this issue is known as crossing the reality gap (Jakobi, 1998; Koos et al., 2010). The *reality gap* is the barrier presented by inevitable discrepancies between a simulated model and its real-world analogue. That is, robot controllers developed in simulation will most likely fail when naively transferred onto a real robot, often because of noise (i.e. non-determinism in sensors and effectors) in the real world.

Most attempts to overcome this problem craft simulations that model the real robot and its environment as accurately as possible (Cliff et al., 1993; Jakobi, 1998; Miglino et al., 1995; Nolfi and Parisi, 1996). It is also common to introduce

non-determinism through noise in the sensors and effectors of the robot in simulation (Cliff et al., 1993; Jakobi, 1998; Koos et al., 2010; Miglino et al., 1995; Nolfi and Floreano, 2000). However, training with noise is not without disadvantages, such as increased computational cost from multiple non-deterministic trials (necessary to counteract variance in fitness measurements) and the difficulty of crafting a sufficiently accurate model with the right distribution of noise. Because of these disadvantages, it would be preferable to train *without* noise if there existed alternatives that also provided robustness. In this spirit, this paper presents a preliminary investigation into the possibility of encouraging robust behaviors using only information from evaluations consisting of a single deterministic trial.

While robustness can only be *verified* over multiple trials, it is still possible that there are clues to robustness hidden within even a single trial. One possible such clue is illuminated by considering how the behaviors of real animals differ from those produced by artificial evolution. Animals are robust because they do not depend upon incidental aspects of the environment (e.g. a herbivore does not depend on a particular configuration of grass blades to feed successfully). However, the same phenomenon does not hold in general for artificial systems; artificial evolution tends to exploit features specific to the simulation not present in reality. Interestingly, observing an animal only once often leaves one with an impression of its robustness. Similarly, an experimenter observing a robot behavior in simulation may often suspect its fragile nature.

The question raised by such impressions is, what cues are being perceived to make such judgments? That is, what are we noticing about animals in nature that makes them seem so vigorous? Perhaps one heuristic for judging robustness is how *reactive* their behavior appears. That is, one clue to robust behavior is noticeably *seeking and reacting* to changes in the perceived environment, which is a trait exhibited widely by natural life. Importantly, by observing a behavior it is possible to estimate how reactive it is. For example, take the behaviors of students during a lecture. If the students nod when key concepts are introduced they are re-

acting appropriately to indicate that they understand; on the other hand, unreactive students with constant blank stares reveal less information. Similarly, a blind man with a cane trying to navigate a corridor often also exhibits reactivity. If the man taps his cane continually against a wall to verify his bearings, the behavior is more reactive than if the man relies completely on a memorized layout of the corridor without re-adjusting (as artificially evolved agents often do). Intuitively, the more reactive tapping behavior would also be more robust to unforeseen changes in the corridor or missteps made by the man. Thus the hypothesis in this paper is that individuals that demonstrate their reactivity by paying attention to the world may generally be stepping stones towards robust behavior. Therefore it may prove effective to directly encourage reactivity, which is the propensity to seek and react to information in the environment continually.

While there may be many ways to quantify the notion of reactivity, the measure in this paper is based on statistical *dependence* between changes in the sensors and the effectors of a robot. Two random variables are dependent if knowing the state of one variable helps predict the other; in other words, there is some relationship between the two variables. In this way, if the magnitude of changes in sensors and effectors of a robot are dependent, it may indicate that the robot is reacting consistently to its environment (i.e. the magnitude of change in environmental input consistently influences the corresponding magnitude of changes in behavior). In this paper such dependence is measured by *mutual information*, which thereby formally captures most closely the informal idea of reactivity introduced here. Indeed, Ay et al. (2008) previously showed an important theoretical connection between maximizing mutual information in sensory experience and effective exploratory behavior in robots. This paper thus suggests how such a measure can be exploited in evolving specific goal-directed behaviors that are resistant to noise.

The idea of incentivizing reactivity to encourage robustness is explored in four maze navigation tasks designed to be challenging under noisy conditions, which makes robustness difficult to achieve. The main result is that rewarding reactivity in single-trial deterministic evaluations *without noise* produces controllers with robustness to noise often rivaling or outperforming those produced by explicitly training with noise. This result is significant because it illuminates that there are hints to robustness observable within a single non-noisy trial, and also establishes a new practical approach to training for robustness, which is a property of general interest both to artificial life and ER.

Background

This section reviews past work in evolving robust controllers in ER, the NEAT and HyperNEAT methods applied in the experiments, and multi-objective optimization.

Evolving for Robustness

For practical reasons, controllers for robots in ER are often trained in a computer simulation rather than directly in reality (Nolfi and Floreano, 2000). However, discrepancies between simulation and reality may cause controllers that are effective in simulation to fail when transferred to a real robot. Because this problem of crossing the reality gap is a significant issue in ER there exist specific training methods that attempt to mitigate it (Bongard and Lipson, 2004; Jakobi, 1998; Koos et al., 2010). The reality gap is one facet of the larger difficulty of evolving general, robust controllers that are not overly dependent on simulation details.

Nearly all training strategies for evolving robust controllers involve training at least some individuals with multiple trials, often non-deterministically (Gomez and Miikkulainen, 2004; Jakobi, 1998; Koos et al., 2010). A common motivation for such training is that real-world sensors often do experience some degree of noise; however, a deeper motivation is that strategically applying noise to a robot's sensors or effectors can prevent evolution from exploiting features specific to a particular simulation (Jakobi, 1998).

While the motivations may be reasonable, the computational cost of training with noise is significant because noisy evaluations normally consist of multiple trials to reduce uncertainty about a policy's average performance (Koos et al., 2010). To reduce computational costs, some methods seek to evaluate only *some* individuals in a full suite of noisy trials by estimating transferability for other individuals (Koos et al., 2010). Yet this approach still requires additional potentially expensive evaluations and the estimates of transferability may not always be accurate. In addition to computational costs, it is not always clear how many trials, in what distribution, and with what intensity noise should be applied in training to ensure successful transfer (Gomez and Miikkulainen, 2004). While Jakobi (1998) lays out a principled methodology based on *minimal simulations*, it still requires painstaking measuring and modeling to implement.

An interesting unexplored question is whether there exist distinguishing properties of robust robot or animat controllers that are visible in a single deterministic trial. If such properties exist and can be explicitly encouraged by an appropriate training incentive, it may be possible to evolve robust robot policies without *any* non-deterministic trials. While interesting in its own right, such a training methodology would also reduce computational cost and the need to model a domain precisely. To this end, the experiments in this paper explore incentivizing the *reactivity* of an evolved controller to encourage its robustness.

Thus these experiments require a method to evolve robot controllers. Though other methods could be applied, here the HyperNEAT neuroevolution method was chosen as a well-established representative method in ER. The next section reviews the Neuroevolution of Augmenting Topologies (NEAT) approach, the foundation of HyperNEAT.

Neuroevolution of Augmenting Topologies

The NEAT method was originally developed to evolve artificial neural networks (ANNs) to solve difficult control tasks (Stanley and Miikkulainen, 2002, 2004). Like the SAGA method (Harvey, 1993) introduced before it, NEAT begins evolution with a population of small, simple networks and *complexifies* the network topology into diverse species over generations, leading to increasingly sophisticated behavior. A similar process of gradually adding new genes has been shown in natural evolution (Martin, 1999).

However, a key feature that distinguishes NEAT from prior work in complexification is its unique approach to maintaining a healthy diversity of complexifying structures simultaneously, as this section reviews. Complete descriptions of the NEAT method, including experiments confirming the contributions of its components, are available in Stanley and Miikkulainen (2002), and Stanley and Miikkulainen (2004). This section briefly reviews the key ideas on which the basic NEAT method is based.

To keep track of which gene is which while new genes are added, a historical marking is uniquely assigned to each new structural component. During crossover, genes with the same historical markings are aligned, producing meaningful offspring efficiently. In traditional implementations of NEAT, speciation protects new structural innovations by reducing competition between differing structures and network complexities, thereby giving newer, more complex structures room to adjust. Networks are assigned to species based on the extent to which they share historical markings. It is important to note that this aspect of NEAT was altered in this paper to replace speciation in NEAT with an explicit genetic diversity objective, which achieves a similar effect. That way, NEAT is easily integrated into a multi-objective framework, as explained shortly. Finally, complexification, which resembles how genes are added over the course of natural evolution (Martin, 1999), is thus supported by both historical markings and protecting innovation, allowing NEAT to establish high-level features early in evolution and then later elaborate on them. In effect, then, NEAT searches for a compact, appropriate network topology by incrementally complexifying existing structure.

The next section reviews HyperNEAT, an extension of NEAT applied in the experiments as a representative example of a modern neuroevolution method.

HyperNEAT

Many neuroevolution methods are *directly encoded*, which means each part in the phenotype is encoded by a single gene, making the discovery of repeating motifs expensive and improbable. Therefore, indirect encodings (Bongard and Pfeifer, 2003; Hornby and Pollack, 2002; Stanley and Miikkulainen, 2003) have become a growing area of interest in evolutionary computation and artificial life.

One such indirect encoding designed explicitly for neural networks is the Hypercube-based NeuroEvolution of Augmenting Topologies (HyperNEAT) approach (Gauci and Stanley, 2010; Stanley et al., 2009), which is an indirect extension of the directly-encoded NEAT approach (Stanley and Miikkulainen, 2002, 2004) reviewed in the last section. This section briefly reviews HyperNEAT; a complete introduction is in Stanley et al. (2009) and Gauci and Stanley (2010). Rather than expressing connection weights as distinct and independent parameters in the genome, HyperNEAT allows them to vary across the phenotype in a regular pattern through an encoding called a *compositional pattern producing network* (CPPN; Stanley, 2007), which is like an ANN but with specially-chosen activation functions.

Such CPPNs are used in HyperNEAT to represent the connectivity patterns of ANNs as a *function of geometry*. That is, if an ANN's nodes are embedded in a geometry, i.e. assigned coordinates within a space, then it is possible to represent its connectivity as a single evolved function of such coordinates. In effect the CPPN paints a pattern of weights across the geometry of a neural network. To understand why this approach is promising, consider that a natural organism's brain is physically embedded within a geometric space, and that such embedding heavily constrains and influences the brain's connectivity. Topographic maps (i.e. ordered projections of sensory or effector systems such as the retina or musculature) exist within brains that preserve geometric relationships between high-dimensional sensor and effector fields (Hubel and Wiesel, 1962; Udin and Fawcett, 1988). In other words, there is important information *implicit* in geometry that can only be exploited by an encoding informed by geometry.

In particular, geometric *regularities* such as symmetry or repetition are pervasive throughout the connectivity of natural brains. To similarly achieve such regularities, CPPNs exploit activation functions that induce regularities in HyperNEAT networks. The general idea is that a CPPN takes as input the geometric coordinates of two nodes embedded in the *substrate*, i.e. an ANN situated in a particular geometry, and outputs the weight of the connection between those two nodes. In this way, a Gaussian activation function by virtue of its symmetry can induce symmetric connectivity and a sine function can induce networks with repeated elements. Note that because CPPN size is decoupled from the size of the substrate, HyperNEAT can compactly encode the connectivity of an arbitrarily large substrate.

It is important to note that HyperNEAT is chosen here simply as a representative modern neuroevolution method. Because all experiments are based on HyperNEAT, the main distinctions among them will be the use of noise or reactivity in training rather than the training algorithm or its particular details. The next section reviews multi-objective optimization, which is combined later with HyperNEAT to enable optimizing both reactivity and fitness during a single run.

Multi-objective Optimization

Multi-objective optimization is a popular paradigm within EC that addresses how to optimize more than one objective at the same time in a principled way (Coello, 1999). The experiments in this paper apply an implementation of NGS-II (Deb et al., 2002), a well-established Pareto-based multi-objective search algorithm, to optimize a traditional fitness objective and a reactivity objective concurrently.

The concept of dominance is central to Pareto-based multi-objective search; the key insight is that when comparing two individuals over multiple objectives, if both individuals are better on different subsets of the objectives then there is no meaningful way to directly rank such individuals because neither entirely *dominates* the other. That is, ranking such mutually non-dominating individuals would require placing priority or weight on one objective at the cost of another; traditionally one individual dominates another only if it is no worse than the other over all objectives and better than the other individual on at least one objective.

In this way, the best individuals in a population are those that are not dominated by any others. Such best individuals form the *non-dominated front*, which defines a series of trade-offs in the objective space. That is, the non-dominated front contains individuals that specialize in various combinations of optimizing the set of all objectives. Some will maximize one at the expense of all the rest, while some may focus equally on all of the objectives. In this way, various tradeoffs of competing objectives such as genomic diversity, fitness, and reactivity can be explored during a single evolutionary run. The hope is that particular trade-offs between fitness performance and reactivity (i.e. policies that perform as well as possible given the constraint that they must be reactive) may lead to more robust behavior.

Recall that a detail of combining NEAT or HyperNEAT with multi-objective optimization is that NEAT has a mechanism (called speciation) for preserving genomic diversity that does not fit naturally into NGS-II. Thus in the experiments in this paper, speciation is replaced in NEAT with an explicit genomic diversity objective that is similar in spirit. In particular, the genomic diversity of a given genome is quantified as the average distance to its k -nearest neighbors in genotype space as measured by NEAT's genomic distance measure. In this way, multi-objective evolution with NEAT is incentivized to maintain genomic diversity in a similar way to how it is in the original formulation of NEAT.

The next section formalizes the measure of reactivity that will be used as an additional objective for training.

Approach: Training for Reactivity

While other measures may also in the future prove effective for encouraging robustness, the hypothesis in this paper is that an agent that is more reactive to its environment may also be more robust. For example, a robot in a maze that is constantly probing and reacting to the walls with its

range-finder sensors as it explores may be more robust than a robot that always executes a memorized plan (which could be disrupted easily by noise). Thus what is needed is a quantification of reactivity that can be directly encouraged during evolution.

In this paper the notion of reactivity is formulated as a measure of statistical dependence between the magnitude of changes in a robot's sensors and its effectors. In general, dependence between two variables implies some kind of relationship between them (e.g. an increase in one variable may tend to result in a decrease in the other). More specifically, it implies that knowledge of one variable helps predict the other. Encouraging such dependence makes sense because it provides evidence that an agent is paying *attention* to changes in its immediate situation. In particular, it implies that the magnitude of change in a robot's sensors influences the magnitude of change of its effectors. In this way, the measure is agnostic to the exact relationship between the two because the ideal such relationship may vary between domains. However, it ensures at least that reactions to sensory changes are consistent, which aligns well with the idea of reactivity.

For example, a particularly attentive student might nod vigorously when a particularly important concept is explained but only slightly when a trivial theorem is proved. However, for the blind man tapping his cane in a corridor, any sudden large change in distance from the wall may call for caution and minor adjustment. Although such a consistent nodding or adjustment policy might not be directly necessary to solve the task, it provides *evidence* that the behavior is reactive. The particular measure of statistical dependence applied here, motivated by Ay et al. (2008), is that of *mutual information* (Shannon, 1949).

The mutual information statistic for two continuous random variables takes the following form:

$$I(X; Y) = \int_Y \int_X p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) dx dy, \quad (1)$$

where $p(x, y)$ is the joint probability distribution function of X and Y , and $p(x)$ and $p(y)$ are the marginal probability distributions of X and Y . The higher the absolute value of $I(X; Y)$, the more dependent are the two variables.

For the experiments in this paper, reactivity is measured by the mutual information between the magnitude of changes in a robot's range-finder sensors and the magnitude of changes in its motor effectors (unlike in Ay et al. (2008), who only measure mutual information in sensors over time). However, this approach is general enough to be applied to different sensory setups in robots in other ER domains where probing and reacting is also important to robustness. Formally, the seven range-finder sensors i_1, \dots, i_7 of the simulated robot are subtracted from their values on the previous timestep and the average magnitude of these differences

at timestep t is recorded as x_t . The average change in the robot's outputs y_t is computed accordingly.

Because the true distributions of X and Y are not known, $p(x)$, $p(y)$, and $p(x, y)$ are estimated through histograms (with a bin width of 0.05) of the sampled data x_t and y_t collected during an evaluation. That is, three histograms are created: two one-dimensional histograms (one over x_t for $p(x)$ and one over y_t for $p(y)$), and one two-dimensional histogram (over both x_t and y_t for $p(x, y)$). Riemann sums are then applied to approximate the integrals from equation 1. However, any reasonable means of estimating the distributions or of numerical integration could be substituted.

While optimizing this formalized measure of reactivity alone would not necessarily lead to successful task performance, it can alternatively be added as an *additional objective* to fitness by employing a multi-objective optimization algorithm. In this way, individuals might be evolved that both solve a given task and provide evidence of potential robustness by being reactive, without multiple noisy trials. The motivation is that if robust solutions could be evolved through this approach, computational costs would be reduced, as would the need for precisely modeling a domain (including appropriate levels of noise).

Maze Navigation Experiments

Because reactivity is intended to encourage robust behaviors, a domain for testing reactivity should be challenging under noisy conditions. Thus four maze navigation domains (figure 1) that create such a challenge in different ways are explored in this paper.

In all of the mazes, a Khepera robot controlled by an ANN must navigate from a starting point to an end point in a fixed time limit that requires direct traversal. The Straight maze (figure 1a) is designed to be simple but incorporate situations that only become *necessary* to experience when an evolved behavior is exposed to significant levels of noise. That is, although an unconditional “always go forwards” policy will be effective without noise, sufficient effector noise may cause the robot's heading to veer into walls. To further accentuate such situations, in this maze the robot is disabled for the remainder of a trial if it collides with a wall. The Zigzag maze (figure 1b) is slightly more complicated because of the need to turn, but it and the remaining mazes allow the robot to recover if it hits a wall. The Winding maze (figure 1c), with its right-angle turns and narrower corridors, creates significant opportunity for the robot to get stuck or confused with increasing noise. Finally, the most challenging maze, the Deceptive maze (figure 1d), has a deceptive cul-de-sac that may complicate training in addition to sharp corners that are difficult to navigate with noise.

The simulated robot is modeled after the Khepera III (K-Team, 2010), and training and testing noise levels are in line with established models of the robot (Cyberbotics, 2012). The robot has six rangefinders that indicate the distance to

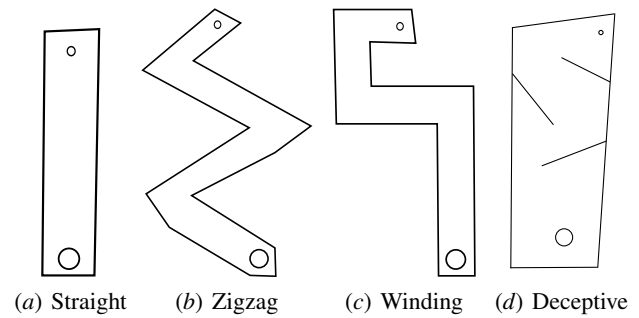


Figure 1: **Domains.** The goal of the agent in the maze navigation domains is to navigate from the starting position (large circle) to the goal (small circle). Note that mazes are not drawn to scale.

the nearest obstacle. Its three effectors produce forces that respectively turn and propel the robot. At each simulated timestep, the robot moves forward at a velocity of $9F$ centimeters per second, where F is the forward effector output. The robot also turns at $120(R - L)$ degrees per second, where R is the right effector output and L is the left effector output. The fitness of an individual is calculated as its distance to the goal at the end of the evaluation, which is a standard measure of progress in maze navigation tasks.

Three different approaches are compared to investigate the potential of training for reactivity:

- In the **Standard** setup there is a single deterministic trial evaluated on two objectives: genomic diversity and the domain-dependent fitness measure.
- In the three **Noise** setups the objectives remain the same as in the standard setup, but each robot is evaluated in *eight* non-deterministic noisy trials to determine its fitness. The amount of both sensor and effector noise for the three different noise setups is respectively 10%, 20%, and 30%, applied as follows: Noise is computed according to the weighted average $(1.0 - x)v + x(n)$, where x is the noise level, v is the before-noise value, and n is randomly chosen from the unit uniform distribution.
- In the **Reactivity** setup an additional reactivity objective (as described earlier) complements the genomic diversity and fitness objectives. As in the Standard setup, the robot is evaluated only in a *single* deterministic trial with no noise.

Experimental Parameters

Because HyperNEAT differs from original NEAT only in its set of activation functions, it uses the same parameters (Stanley and Miikkulainen, 2002). The experiments were run with a modified version of the public domain SharpNEAT package (Green, 2006). The size of each population was 250 with 20% elitism. Asexual offspring (50%) had 0.96 probability of link weight mutation, 0.03 chance of link addition, and 0.01 chance of node addition. The coefficients

for determining genomic similarity were 1.0 for nodes and connections and 0.1 for weights. The available CPPN activation functions were sigmoid, Gaussian, absolute value, and sine. Parameter settings are based on standard SharpNEAT defaults and prior reported settings for NEAT (Stanley and Miikkulainen, 2002, 2004). They were found to be robust to moderate variation through preliminary experimentation. Runs of the Straight, Zigzag, and Winding mazes lasted 400 generations, while because of its increased difficulty runs of the Deceptive maze lasted 1,000 generations.

Results

In *training*, the Reactivity setup did not significantly differ in performance from the other setups in the Straight or Winding mazes. However, the Reactivity setup did solve the Deceptive maze more often (in 17 out of 20 runs) than any other setup (Fisher's exact test; $p < 0.001$). In comparison, the Standard setup solved the maze in 8 runs, and the 10%, 20%, and 30% Noise setups solved the maze in 3, 1, and 0 runs, respectively. The Reactivity setup also solved the Zigzag maze significantly more often than the 20% or 30% Noise setups (Fisher's exact test; $p < 0.001$). These results support the hypothesis noise may often complicate training. However, training performance may not reflect robustness to noise; the Standard and Reactivity setups in fact both had no exposure to noise at all. It is important to note that even when a complete solution is not evolved in training, a partial evolved solution might still sometimes solve the task in the more lenient generalization test that is described next.

Because the motivation for this paper is to investigate the robustness of evolved controllers, a generalization test was devised to measure how well an evolved controller would perform in noisy distributions not encountered during training. The generalization test consisted of 50 noisy trials with the length of evaluation doubled from training to allow for greater leniency. Such leniency reflects that in transfer slight stumbles due to the reality gap are preferred to catastrophic failure (i.e. if a policy will never solve the task irrespective of how much time is allotted). An individual receives a score on the generalization test in accordance with the fraction of trials in which it is able to navigate the maze successfully (i.e. if it comes within 20 units of the goal at any time). For each run, the individual scoring the overall highest on this test from sampling the population every 100 generations is recorded (except in the Deceptive maze experiment in which every 200 generations is recorded because of its longer duration), and averaged over each of the 20 runs. This approach to testing gives a sense of the most robust controller one can hope to find with each approach. The generalization test is repeated with noise distributions from 0% to 35% at 5% intervals. Thus over five setups (three training levels of noise, standard, and reactivity) with eight testing noise levels each, there are 40 total generalization scenarios per domain, and 32 possible pairwise comparisons between Reactivity and

the other setups in each domain. The results of applying this generalization test are shown in figure 2.

To assess statistical significance on the generalization test for each domain, a one-way ANOVA test was first applied across the five experimental setups for each level of generalization noise to demonstrate that the distributions are significantly different (at least $p < 0.05$). If at a particular noise level this first test was passed, then Student's t-tests were applied to measure the significance of pairwise differences between Reactivity and the other experimental setups.

The Straight maze, as might be expected, proved challenging only to the Standard setup because this setup provided no incentive to learn to interact with walls. Supporting its motivation, the Reactivity setup, despite not being exposed to noise nonetheless discovers policies that robustly react to walls. There were only two significant differences (among 32 total pairwise comparisons) between Reactivity and the other setups in the Zigzag maze (Reactivity was better than Standard in one scenario and 30% Noise was better than Reactivity in another), indicating perhaps that in some relatively simple domains it may make little difference what training setup is chosen. In the Winding maze, training with higher levels (20% or 30%) of noise provided a significant advantage over Reactivity for generalization with higher levels of noise ($\geq 25\%$), demonstrating that sometimes knowing the distribution of noise in reality can inform training. Finally, the Deceptive maze proved the most challenging for all methods (no method scored above 50% success on the highest noise level in the generalization test), although Reactivity was significantly better than the 30% or 20% Noise setups when testing generalization on low levels of noise ($< 15\%$). This result suggests that an inaccurate noise model can hurt noisy training while reactivity can sometimes circumvent the need for such modeling entirely.

Over all four domains, training with the Reactivity setup was never significantly worse at generalizing than training with the Standard setup, and was significantly better in 15 out of the 32 pairwise comparisons. Training with the Reactivity setup was significantly better at generalizing than the Noise setups in 7 out of 96 comparisons while Noise also was significantly better than Reactivity in 7 pairwise comparisons. Interestingly, the occasional significant advantages for the Noise setups only occurred when the noise level in the generalization test was 25% or greater, which suggests that reactivity training may generally be most advantageous when dealing with moderate levels of noise.

Discussion

The motivation for reactivity is to encourage an agent to *pay attention* to its environment and thereby make full use of its sensory experience. While ultimately the most reactive solution may not be the best performing or most robust, such reactivity may still be desirable because it can potentially act as a stepping stone on the way to a robust policy.

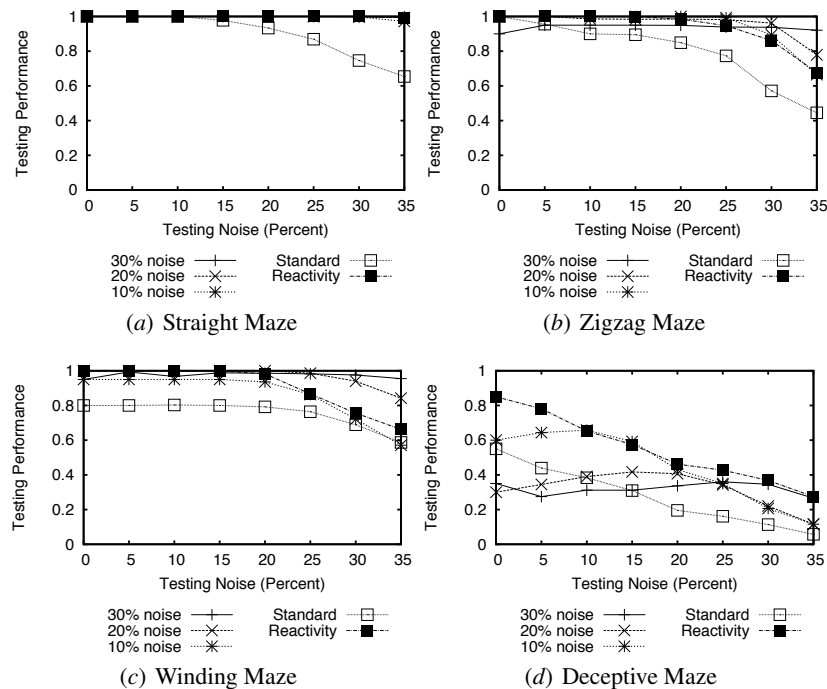


Figure 2: **Maze Navigation Generalization Test Results.** The average probability of the best individual from a run to solve the generalization test at various levels of noise is shown for different training methodologies over the four maze domains. The main result is that training with reactivity in all four domains is never significantly worse than training with noise (10%, 20%, or 30%) on the generalization test at moderate levels of noise ($< 25\%$).

The experiments in this paper provide evidence for this idea because the Reactivity setup often significantly outperforms the Standard setup in generalization testing and never underperforms it, meaning that simply encouraging an agent to be reactive often promotes robust behaviors. Additionally, reactivity is always at least as good at generalizing as the Noise setups when exposed to moderate noise, and in some cases is significantly better. Thus reactivity demonstrates that it is possible to evolve controllers that perform well in noisy situations without ever exposing those controllers to noise. In addition to providing a compelling proof of concept, reactivity also can reduce computational cost (i.e. it took eight times fewer trials per evaluation than noisy training) and the need to model a domain precisely.

One major benefit of training with reactivity is that training with noise requires several noisy trials to be run per evaluation to evaluate a behavior effectively, while reactivity can be accurately measured with a single deterministic trial. Computing an agent's reactivity does require calculating a statistical measure, but this cost is generally insignificant when compared to the computation required to simulate a domain. So even when reactivity does not outperform noisy training, it may still be preferable because of the decreased runtime. Additionally, reactivity can facilitate training robots in complex domains in which the computational costs incurred by multiple, noisy trials are prohibitive.

Another benefit of reactivity is that it can reduce the need

for precise domain models. Accurately modeling a robot, its environment, and the actual levels of sensor and effector noise is often a difficult and laborious task, and perfect accuracy is generally impossible (Jakobi, 1998). However, with noisy training model accuracy can be important; selection of the right level of training noise is necessary to outperform reactivity in the Winding maze or to avoid underperforming reactivity in the Deceptive maze. Thus when training with noise, unless the model is accurate, generalization performance may be suboptimal. Interestingly, the Reactivity setup does not require a model of noise and performance degrades gracefully as the amount of noise increases. Even without any exposure to noise it is rarely significantly worse than any of the Noise setups; in as many cases it is significantly better. Thus it is possible to exploit reactivity to avoid crafting an accurate noise model, which is oftentimes difficult or time-consuming. In future work evolved reactive behaviors will be transferred to the real world to verify these potential benefits for crossing the reality gap.

While training with noise has established itself as the dominant means of producing robust controllers (Gomez and Miikkulainen, 2004; Jakobi, 1998; Koos et al., 2010; Nolfi and Floreano, 2000), the effort required to produce an accurate noise model and the computational cost of training with noise make it a kind of "necessary evil" for real-world transfer. The preliminary results in this paper demonstrate that reactivity provides an alternative to training with

noise that offers performance gains and reduced computational cost in some cases. However, there are still significant avenues for future research in this area. First, the measure of reactivity expressed in this paper is simple and intuitive: The magnitude of the change in outputs should depend on the magnitude of the change in inputs. However, more sophisticated or domain-dependent properties of evolved behaviors may exist that better encourage robustness. Additionally, reactivity could be *combined* with noisy training to further boost performance by encouraging controllers to react appropriately in noisy environments. Ultimately the results in this paper highlight that the idea of rewarding reactivity or other behavioral properties indicative of robustness is a promising research direction that merits further study.

Conclusion

This paper introduced the idea of encouraging properties of evolved controllers observable in single deterministic evaluations that correlate with increased robustness and generality. Motivated by the insight that robust behaviors tend to probe and react to their environment, the reactivity of a controller is suggested as one promising such property. Experiments showed that training with reactivity most often performs as well as training explicitly with noise, and is also significantly better as often as it is worse. The benefit is the reduced computation from considering only one deterministic evaluation and the eliminated need for accurate noise models. While the investigated measure does not always outperform training with noise, it is interesting and counterintuitive that even sometimes training without noise can be more effective in the face of noise than explicitly training with it. The conclusion is that reactivity is a viable new perspective on training for robustness that demonstrates that there may often be hints to robustness or generality hidden within single trials.

Acknowledgements

This research was supported by DARPA and ARO through DARPA grant N11AP20003 (Computer Science Study Group Phase 3), and US Army Research Office grant Award No. W911NF-11-1-0489. This paper does not necessarily reflect the position or policy of the government, and no official endorsement should be inferred.

References

Ay, N., Bertschinger, N., Der, R., Güttler, F., and Olbrich, E. (2008). Predictive information and explorative behavior of autonomous robots. *The European Physical Journal B*, 63(3):329–339.

Bongard, J. and Lipson, H. (2004). Once more unto the breach: Co-evolving a robot and its simulator. In *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9)*, pages 57–62.

Bongard, J. C. and Pfeifer, R. (2003). *Evolving complete agents using artificial ontogeny*, pages 237–258. Morpho-functional Machines: The New Species (Designing Embodied Intelligence). Springer-Verlag.

Cliff, D., Husbands, P., and Harvey, I. (1993). Evolving visually guided robots. In *Proceedings of the second intl. conf. on sim. of adaptive behavior*, pages 374–383. MIT Press.

Coello, C. (1999). A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information systems*, 1(3):129–156.

Cyberbotics (2012). Webots. Commercial Mobile Robot Simulation Software.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197.

Gauci, J. and Stanley, K. O. (2010). Autonomous evolution of topographic regularities in artificial neural networks. *Neural Computation*, page 38. To appear.

Gomez, F. and Miikkulainen, R. (2004). Transfer of neuroevolved controllers in unstable domains. *Lecture Notes in Computer Science*, pages 957–968.

Green, C. (2003–2006). SharpNEAT homepage. <http://sharpneat.sourceforge.net/>.

Harvey, I. (1993). *The Artificial Evolution of Adaptive Behavior*. PhD thesis, School of Cognitive and Computing Sciences, University of Sussex, Sussex.

Hornby, G. S. and Pollack, J. B. (2002). Creating high-level components with a generative representation for body-brain evolution. *Artificial Life*, 8(3):223–246.

Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *The Journal of Physiology*, 160:106–154.

Jakobi, N. (1998). Minimal simulations for evolutionary robotics. Technical report.

K-Team (2010). Khepera III mobile robot. <http://www.k-team.com>.

Koos, S., Mouret, J., and Doncieux, S. (2010). Crossing the reality gap in evolutionary robotics by promoting transferable controllers. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 119–126. ACM.

Martin, A. P. (1999). Increasing genomic complexity by gene duplication and the origin of vertebrates. *The American Naturalist*, 154(2):111–128.

Migilino, O., Lund, H., and Nolfi, S. (1995). Evolving mobile robots in simulated and real environments. *Artificial life*, 2(4):417–434.

Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics*. MIT Press, Cambridge.

Nolfi, S. and Parisi, D. (1996). Learning to adapt to changing environments in evolving neural networks. *Adaptive behavior*, 5(1):75–98.

Shannon, C. (1949). A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423.

Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162.

Stanley, K. O., D’Ambrosio, D. B., and Gauci, J. (2009). A hypercube-based indirect encoding for evolving large-scale neural networks. *Artificial Life*, 15(2).

Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127.

Stanley, K. O. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.

Stanley, K. O. and Miikkulainen, R. (2004). Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21(1):63–100.

Udin, S. and Fawcett, J. (1988). Formation of topographic maps. *Annual review of neuroscience*, 11(1):289–327.

Aracna: An Open-Source Quadruped Platform for Evolutionary Robotics

Sara Lohmann*, Jason Yosinski*, Eric Gold, Jeff Clune, Jeremy Blum and Hod Lipson

Cornell University, 239 Upson Hall, Ithaca, NY 14853
sml253@cornell.edu, yosinski@cs.cornell.edu

Abstract

We describe a new, quadruped robot platform, Aracna, which requires non-intuitive motor commands in order to locomote and thus provides an interesting challenge for gait learning algorithms, such as those frequently developed in the Evolutionary Computation and Artificial Life communities. Aracna is an open-source hardware project composed of off-the-shelf and 3D-printed parts, enabling other research teams to modify its design according to their scientific needs. Aracna was designed to overcome the shortcomings of a previous quadruped robot platform, whose legs were so heavy that the motors could not reliably execute the commands sent to them. We avoid this problem by locating all motors in the body core instead of on the legs and through a design which enables the servos to have a greater mechanical advantage. Specifically, each of the four legs has two joints controlled by separate four-bar linkage mechanisms that drive the pitch of the hip joint and knee joint. This novel design causes unconventional kinematics, creating an opportunity for gait-learning algorithms, which excel in counter-intuitive design spaces where human engineers tend to underperform. Because it is low-cost, flexible, kinematically interesting, and an improvement over a previous design, Aracna provides a useful new hardware platform for testing algorithms that automatically generate robotic behaviors.

Introduction

There is a long history in the Artificial Life and Evolutionary Robotics community of automatically generating behaviors for robots (Nolfi and Floreano, 2000; Pfeifer et al., 2007; Sims, 1994; Hornby et al., 2005; Lipson and Pollack, 2000). Much work has focused on evolving gaits for legged robots (Clune et al., 2009, 2011; Hornby et al., 2005, 2003; Kodjabachian and Meyer, 1998; Koos et al., 2011; Bongard et al., 2006; Yosinski et al., 2011; Gallagher et al., 1996). While some of this previous work involved evolution directly on a physical robot (Yosinski et al., 2011; Zykov et al., 2004), more often a gait was evolved in simulation and then transferred to the physical robot (Lipson et al., 2006; Koos et al., 2011; Hornby et al., 2005; Bongard et al., 2006). Many of these studies report that evolutionary algorithms

produced gaits that outperformed those designed by a human engineer (Yosinski et al., 2011; Hornby et al., 2005), which is not surprising given that evolutionary algorithms routinely create solutions that are superior to manually created solutions (Koza, 2003).



Figure 1: Aracna: an open-source 3D printed quadruped robot platform, here printed in a black rapid prototyping polymer. STL files for 3D printing the robot and drivers for the Arbotix board and servos are publicly available at <http://creativemachines.cornell.edu/aracna>.

The results just mentioned suggest that evolutionary algorithms are a promising approach for generating gaits and other behaviors for physical robots. Despite this promise, the field remains small, partly because robots are expensive, and they are difficult to modify. Access to cheap, customizable robots could increase the number of researchers able to participate in the field. Moreover, in nearly all of the papers mentioned previously, the robots were custom-made, preventing teams at other universities from reproducing the results of other groups and or testing new algorithms on a robotic platform used in a previous study. That, in turn, slows the progress of science because it is difficult to interpret whether the variance in results between different studies

* Authors contributed equally to this work.

was due to the algorithms used or the robotic platform those algorithms were tested on.

Some robot platforms are emerging, but they tend to be wheeled robots without complex kinematics, such as the ePuck (Mondada et al., 2009). Wheeled robots are interesting testbeds for many robotic behaviors, but they do not allow gait evolution and are unable to traverse rugged terrains. Legged robotic platforms exist, but they tend to be extremely expensive, such as the Aldebaran Nao, which costs more than \$10,000 USD. Another drawback to these commercial platforms is that it is hard, if not impossible, to modify the hardware design because they are not open-source hardware projects, and do not take advantage of off-the-shelf components and 3D printing, meaning that complex manufacturing tools are required to manufacture newly designed parts.

In this paper we address these needs by introducing *Aracna*, a low-cost, open-source, easily customizable robot platform with non-intuitive walking kinematics (Figure 1). *Aracna* is the third quadruped robot developed for evolutionary learning algorithms by the Creative Machines Lab at Cornell University (Bongard et al., 2006; Yosinski et al., 2011). Like the most recent of the two previous designs, called *QuadraTot* (Yosinski et al., 2011), the body of *Aracna* is 3D printed and the STL files are available online, meaning that other researchers can easily customize the body's design. As in the both previous designs, each leg has a hip and knee joint controlled by two actuators. The original Creative Machines Lab quadruped robot favored starfish-like movements (Bongard et al., 2006). The second quadruped robot — *QuadraTot* — developed spider-like movements, but was found to be limited by its weight and lack of mechanical advantage, such that the motors would overheat and time-out when trying to execute many commands (Yosinski et al., 2011; Glette et al., 2012). We designed *Aracna* to be able to produce fast, spider-like movements, yet be lightweight enough that the motors would not overheat. We also designed *Aracna* to be inexpensive: as described below, its overall price is under \$1,400 USD. In the following sections we describe the *Aracna* platform in more detail.

Overall Hardware Design

The hardware of *Aracna* was designed to improve upon the previous Creative Machines Lab quadruped robots (Bongard et al., 2006; Yosinski et al., 2011), while still qualitatively resembling those robots. *Aracna* is similar in that it has a body and four legs, with each leg having two joints that can pitch forward and back like knees (Figure 2 and 3).

One change was to constrain the movement of the joints toward the goal of creating faster, spider-like movements. To prevent starfish-like movements and instead encourage a walking gait with the robot body permanently off the ground, the legs were constrained such that they cannot straighten out and the knee cannot hyperextend.

Another change was to reduce the both the overall weight

of the robot and the weight of each leg. Two previous studies that used the *QuadraTot* robot report that the motors quickly wore out and could not reliably execute the commands sent to them, likely because of both the overall weight of the *QuadraTot* and the fact that housing servos on the legs made them heavy (Bongard et al., 2006; Yosinski et al., 2011). The weight of the robot's core was reduced in a number of ways.

Initially, we eliminated the *QuadraTot*'s fit-PC, an on-board Linux computer weighing 370g, and replaced it with an on-board *ArbotiX* microcontroller that weighs only 47g. Wireless communication between the external control computer and the *ArbotiX* microcontroller occurs over wireless XBee.

A second means of eliminating weight involved switching to a lighter battery. The *QuadraTot* had two 12V lithium-ion battery packs that weighted 140g each for a total of 280g. *Aracna* has a single lithium-polymer 11.1V battery that weighs 122 g. As with the *QuadraTot*, *Aracna* can also run tethered to power, if desired, to avoid the need to run from battery power. This may be helpful for extended experiments.

A major modification, targeted at reducing the weight of legs, was the use of two four-bar mechanisms to drive the joints in each leg. This mechanism causes the controlled joint to move at a fraction of the output angle of the actuator, giving the motor a relatively larger mechanical advantage over the position of each leg. Figure 3 shows the crank-rocker system, where the input crank is actuated by a servo and the rocker is the leg. This configuration allows the servo motors to be contained in the robot core, reducing both the inertia and mass of each leg. The weight of an *Aracna* leg is 105g compared to the 217g for a *QuadraTot* leg.

Combined, these changes to minimize weight led to a 31.4 percent reduction in weight of the robot. The *QuadraTot* weighs 1.88kg whereas *Aracna* weighs 1.29kg.

A final change was to upgrade the power of the servo motors in order to increase the ability of the robot to strike whichever configurations are specified by the learning algorithms. Specifically, we upgraded from Dynamixel AX 12+ motors to AX-18A motors, which have a higher stall torque (1.8Nm vs. 1.5Nm at 12V), a higher stall current (2.2A vs 1.5A), and a higher no-load speed (97 vs. 59 RPM).

3D Printed Body

The body of *Aracna* takes advantage of 3D printing technology, also known as additive manufacturing, which generates physical objects from digital designs by building them up layer by layer (Gibson et al., 2009; Lipson and Kurman, 2010). The use of 3D printing means that other *Aracna* users can easily make copies of *Aracna*, either by having access to a 3D printer or via online 3D printing services such as Shapeways, Sculpteo, or other online vendors. Either option requires the 3D design files in the stereolithography (STL) format, which are published in the online support material

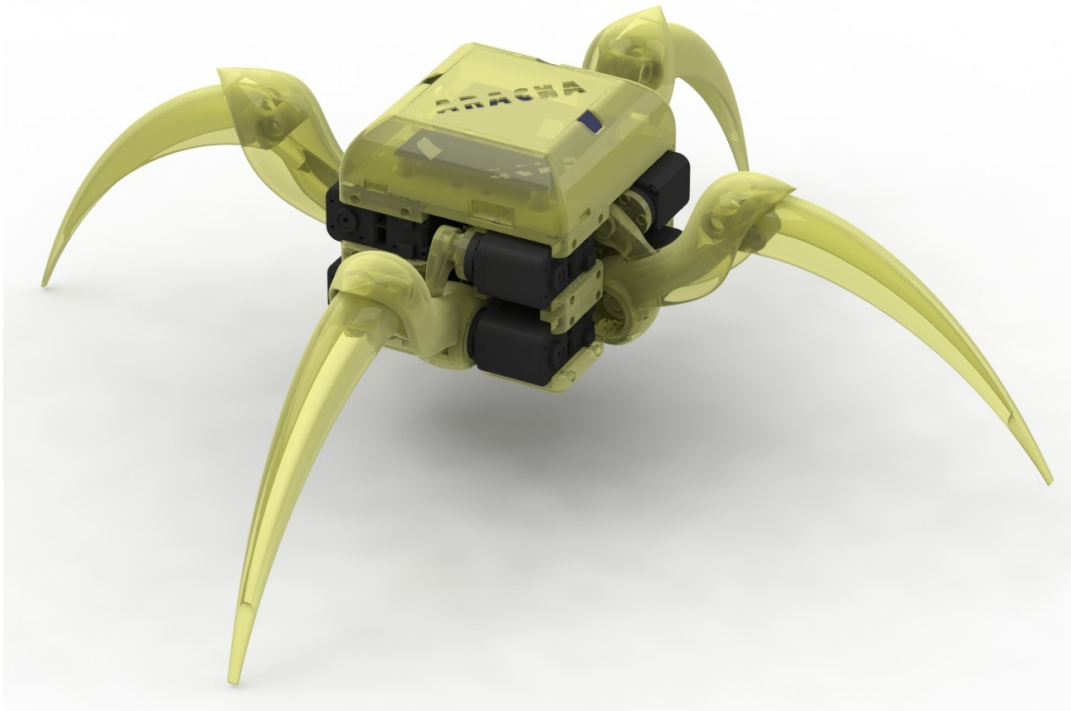


Figure 2: A rendered CAD model of Aracna. Note the lack of heavy servos on the legs themselves, which are instead controlled via four-bar linkages by servos in the robot's core.

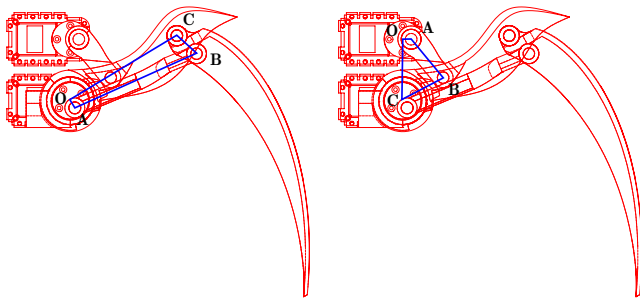


Figure 3: Crank-rocker four-bar linkage controlling flexion/extension of the knee and hip joints. In both cases above, the input crank (link OA) is actuated by a servo, the rocker is the leg (link CB), and the fixed link is OC.

for this paper (Aracna, 2012). Moreover, to catalyze innovation in this open-source hardware project, we are also providing the source files for the SolidWorks computer-aided-design program, to enable others to modify the design. It is thus possible for future Aracna users to improve or alter the design and quickly obtain a physical instantiation of the design. Importantly, the use of 3D printing eliminates the need to know how to machine parts, allowing many more researchers to participate in using physical robot morphologies that they design themselves. These ideas are in line with a broader trend toward enabling non-technical users to design and manufacture physical objects (Clune and Lipson, 2011; Clune et al., 2013; Lipson and Kurman, 2010).

An initial version of Aracna was designed to be printed in one piece (Figure 5). However, if one part of the robot became damaged, an entire new robot had to be reprinted, which took over 26 hours and costs roughly \$355 USD on an Objet Connex500 printer. To make repairing the robot easier, cheaper, and quicker, Aracna was redesigned to be modular. It consists of 15 pieces—four legs and the core—that can be separately 3D printed (Figure 6). Printing a leg takes 3.3 hours and costs roughly \$64 USD. Printing the core takes 3 hours and costs \$101 USD. All 15 Aracna pieces can still be printed as one print job, with an overall time of approximately 10 hours and cost of \$308. These figures are based

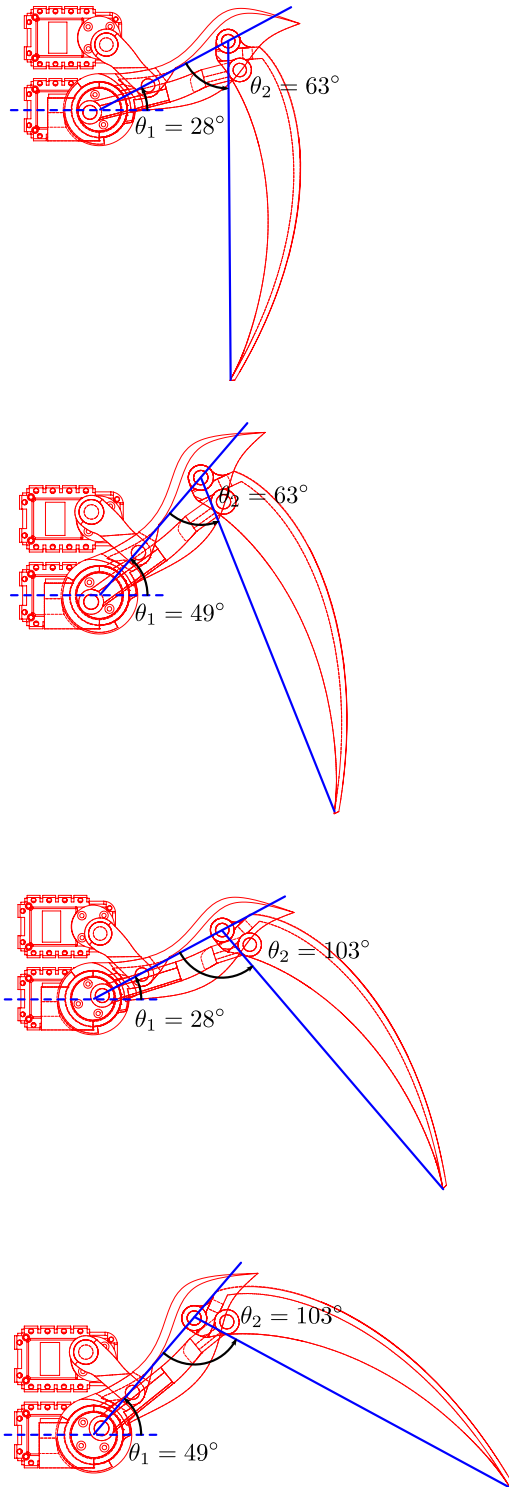


Figure 4: The range of motion of the hip and knee joints in each leg of Aracna. The hip joint rotates by 21.3° and the knee joint by 40.2° .

on Aracna's use of approximately 967g of model material and 746g of support material, and current material costs of 4.5g per USD and 8g per USD for rigid and support material, respectively. This cost estimate is variable depending on the type of material used. The print times are estimates calculated by the Objet's software and are meant to be used as a relative comparison of print times. Table 1 outlines the total estimated cost of Aracna, which is just under \$1,400, including its off-the-shelf electronic components.

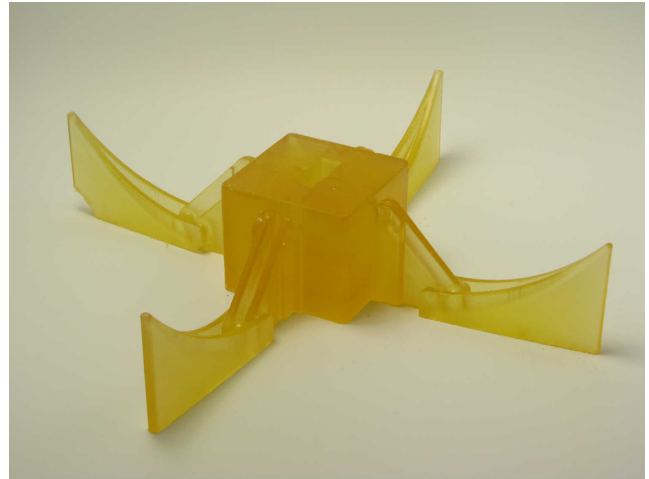


Figure 5: A draft version of Aracna that was printed in one piece. This monolithic design proved expensive to maintain if part of the robot was damaged, and was replaced in a later version with a modular design. This figure shows the printed body with support material still present.

Part	Cost
3D Print Materials	\$308
ArbotiX Robocontroller Kit	\$189
Dynamixel AX-18A Robot Actuator (x8)	\$721
3S 11.1V 2000mAh Pro Lite LiPo Battery	\$73
LiPo Battery Balance Charger Kit	\$70
Cables, Connectors, Misc	\$28
Total	\$1389

Table 1: Estimated total cost. The cost of components and printing material reflect market prices from March 2012. A complete parts list is on our website (Aracna, 2012).

Control

In addition to reducing the weight of the legs, the four-bar mechanisms also satisfied the design goal of making a robot that had non-traditional movements. Unusual kinematics make for a more effective algorithmic test platform, since gait learning algorithms are most helpful in domains



Figure 6: A final version of Aracna printed in multiple pieces. The body is printed as two pieces, with 9 smaller parts within the top piece to reduce support material and print time. These parts can be printed individually if replacements are necessary. The body consists of a total of 11 parts. This image shows a set of 12 printed parts (the complete body and a single leg) with support material still present.



Figure 7: Aracna with optional top cover.

that humans find hard to program solutions for. The reason Aracna's kinematics are counter-intuitive is because there is a nonlinear mapping between each servo's output and the movement of the joint controlled by that servo.

The range of motion for the hip joint is 21.3° and that of the knee joint is 40.2° . These ranges are realized over a servo motion of 184° and 192° for the hip and knee joints, respectively. These amounts are notably larger than the corresponding joint motions, which produces the desired mechanical advantage.

There are two paradigms for encoding movements for Aracna, increasing its flexibility as a testing platform. The first method is to specify explicitly a sequence of positions over time for all eight servos. The second method is to set the speed of each servo. This method is possible because with the four bar mechanism a servo constantly rotating in one direction will move the joint back and forth between its minimum and maximum opening angle. This latter method provides a much smaller search space and would encourage regular gaits, which have been shown to be beneficial when evolving gaits for legged robots (Clune et al., 2011; Hornby et al., 2005).

Software

The software, which is also open-source and freely available (Aracna, 2012), is written in Python and based on the code developed for the QuadraTot platform (Yosinski et al., 2011). The software translates a series of requested joint angles from the learning algorithm into servo movements. Additionally, it returns information to the learning algorithm, such as the distance traveled or the specific trajectory the robot took, so the learning algorithm can assess the quality of the gait. To provide this information, an infrared light emitting diode (LED) was placed on the robot and a Nintendo Wii remote was attached overhead. The software uses the combination of the two to ascertain the X, Y position of the robot. The software is interoperable with any gait or behavior learning algorithm.

Example Gaits

Evolutionary algorithms work best when they have a gradient to follow through a space rich with partial solutions. To get a sense of how randomly-generated gaits would perform, we chose a few gaits by setting random positions and having the robot interpolate between them in a repeated pattern. We found that many such patterns resulted in motion. Videos of several gaits are available on the website (Aracna, 2012).

Conclusion

Here we have introduced Aracna, a low-cost open source platform for evolutionary robotics. The complex kinematics along with the open source nature of the robot will provide an interesting and challenging platform for comparing walking gait algorithms. The updated platform is modular, allow-

ing for low-cost replacement parts and varied leg designs. Future work can include modifying a single leg to have different linkage lengths, or to replace a kinematic joint with a compliant, or flexible, joint. Aracna will enable multiple users to compare data across a single lightweight, low-cost evolutionary robotic platform.

Acknowledgements

This work was supported by the National Science Foundation's Office of Emerging Frontiers in Research and Innovation (grant number 0735953) and an NSF Postdoctoral Research Fellowship in Biology to Jeff Clune (DBI-1003220).

References

- Aracna (2012). Aracna open-source quadruped robot. Open-source hardware and software files are available at <http://creativemachines.cornell.edu/aracna>.
- Bongard, J., Zykov, V., and Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science*, 314(5802):1118–1121.
- Clune, J., Beckmann, B., Ofria, C., and Pennock, R. (2009). Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 2764–2771.
- Clune, J. and Lipson, H. (2011). Evolving three-dimensional objects with a generative encoding inspired by developmental biology. In *Proceedings of the European Conference on Artificial Life*, pages 144–148.
- Clune, J., Stanley, K., Pennock, R., and Ofria, C. (2011). On the performance of indirect encoding across the continuum of regularity. *IEEE Transactions on Evolutionary Computation*, 15(4):346–367.
- Clune, J., Yosinski, J., Doan, E., and Lipson, H. (2013). Automating the design of physical objects via interactive, crowd-sourced evolution based on concepts from developmental biology. *In preparation*.
- Gallagher, J., Beer, R., Espenschied, K., and Quinn, R. (1996). Application of evolved locomotion controllers to a hexapod robot. *Robotics and Autonomous Systems*, 19(1):95–103.
- Gibson, I., Rosen, D., and Stucker, B. (2009). *Additive manufacturing technologies: rapid prototyping to direct digital manufacturing*. Springer Verlag.
- Glette, K., Klaus, G., Zagal, J. C., and Torresen, J. (2012). Evolution of locomotion in a simulated quadruped robot and transfer to reality. In *Proceedings of the Seventeenth International Symposium on Artificial Life and Robotics*.
- Hornby, G., Lipson, H., and Pollack, J. (2003). Generative representations for the automated design of modular physical robots. *IEEE Transactions on Robotics and Automation*, 19(4):703–719.
- Hornby, G., Takamura, S., Yamamoto, T., and Fujita, M. (2005). Autonomous evolution of dynamic gaits with two quadruped robots. *IEEE Transactions on Robotics*, 21(3):402–410.
- Kodjabachian, J. and Meyer, J. (1998). Evolution and development of neural controllers for locomotion, gradient-following, and obstacle-avoidance in artificial insects. *IEEE Transactions on Neural Networks*, 9(5):796–812.
- Koos, S., Mouret, J., and Doncieux, S. (2011). The Transferability Approach: Crossing the Reality Gap in Evolutionary Robotics. *IEEE Transactions on Evolutionary Computation*, pages 1–25.
- Koza, J. (2003). *Genetic programming IV: Routine human-competitive machine intelligence*. Kluwer.
- Lipson, H., Bongard, J., Zykov, V., and Malone, E. (2006). Evolutionary robotics for legged machines: from simulation to physical reality. In *Intelligent Autonomous Systems*, volume 9, page 9.
- Lipson, H. and Kurman, M. (2010). Factory@ home: The emerging economy of personal manufacturing. *Papers in Science and Technology Policy*.
- Lipson, H. and Pollack, J. (2000). Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974–978.
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klap-tocz, A., Magnenat, S., Zufferey, J., Floreano, D., and Martinioli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions*, volume 1, pages 59–65.
- Nolfi, S. and Floreano, D. (2000). *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*. MIT Press, Cambridge, MA.
- Pfeifer, R., Bongard, J., and Grand, S. (2007). *How the body shapes the way we think: a new view of intelligence*. The MIT Press.
- Sims, K. (1994). Evolving 3D morphology and behavior by competition. *Artificial Life*, 1(4):353–372.
- Yosinski, J., Clune, J., Hidalgo, D., Nguyen, S., Zagal, J., and Lipson, H. (2011). Evolving robot gaits in hardware: the hyperneat generative encoding vs. parameter optimization. In *Proceedings of the European Conference on Artificial Life*, pages 890–897.
- Zykov, V., Bongard, J., and Lipson, H. (2004). Evolving dynamic gaits on a physical robot. In *Proceedings of Genetic and Evolutionary Computation Conference, Late Breaking Paper, GECCO*, volume 4.

Analysis of Evolved Agents Performing Referential Communication

Santosh Manicka

Center for Complex Networks and Systems,
Indiana University Bloomington
smanicka@indiana.edu

Abstract

A pair of Continuous-time Recurrent Neural Network (CTRNN) based agents called “Sender” and “Receiver” is evolved on a circular world. Their collective objective is to communicate and move to a target – the Sender needs to communicate the address of a target location on the circle, and the Receiver needs to move to that location after receiving the communication. In extension of previous work (Williams and Beer, 2008), the agents are evolved under conditions different from the original work. Qualitative analysis of the most successful agent-pair shows that the Receiver’s behavior is reminiscent of Newton’s equations of motion in relating its initial velocity to the target address communicated to it. Further analysis using information-theoretic tools reveals a pair of neurons that hold crucial information required for the successful functioning of the Receiver. They are also shown to employ the same kind of information for slightly different purposes.

Background

An act of “referential communication” refers to a meaningful exchange of signals that “point to states of affairs removed in space and/or time” (Williams and Beer, 2008). Nature abounds with such communication, between humans, for example, when we *refer* to things during conversations. A typical example in the simpler invertebrate world is the waggle dance of the bees. A forager bee explores its environment for food sources and once it finds them, returns to the hive and communicates the location of the source to other bees in a characteristic dance. Quite a few models have been developed to study this behavior in bees (see references in Williams and Beer, 2008). Many of them start with the assumption of certain explicit signals that represent things in the environment. The model that Williams and Beer (2008) have developed argued against such approaches, in favor of a model in which signals are evolved. Their model consists of a *Sender* and a *Receiver* that coexist on a circular world. The task is to have the Sender communicate a certain target location to the Receiver and to have it successfully commute to that location. No assumptions are made about any *coding* system *shared* between the Sender and the Receiver that will help evolve the communication process. Instead, the Sender only has sensors through which it receives information and sensors through which it detects the proximity of the Receiver. Their evolved Senders “nudged” the Receivers in a characteristic way that that acted as a code for the information

it transmitted. The point was that sensors and actuators could be *evolved* to be used for symbolic communication.

In this work, we setup a very similar evolutionary experiment and then analyze the evolved Sender-Receiver pair that solves the task successfully. Our main goal is to gain some operational insight into these agents so as to understand one possible way of referential communication better. In the original work, the earliest agent-pairs had evolved “shepherding” strategy where the Sender literally holds the hands of the Receiver until they both reach the target, or “sit and wait” strategy where the Sender is mostly stationary while bouncing around the Receiver and communicating when they meet. Later they modified the experimental setup to have something called a “constraint zone” within which the movement of the Sender was restricted, thus making the Sender communicate about target locations, to which it is spatially and temporally separated (in the true essence of referential communication). Successful solutions evolved under these conditions involved multiple Sender-Receiver interactions. Even though the setup was simple in their experiments, their evolved solutions were somewhat complex in that both the Sender and Receiver could move simultaneously for a certain period of time. In such cases, it would be difficult to understand exactly how the Sender was transmitting information to the Receiver. To overcome such difficulties, we made several modifications to their original setup. Mainly, we split the experiment into two clear phases, one of which is a *communication* phase. During this phase, only the Sender is allowed to move and the Receiver is clamped. During the following *courier* phase, the Sender is removed from the environment and the Receiver is allowed to move. These modifications have somewhat simplified the analyses. We were also able to evolve successful solutions to the problem with this new setup. We have analyzed the mechanisms of the evolved agents mainly using information-theoretic tools.

Methods

A pair of agents, namely “Sender” and “Receiver” lives on a circular world. Any point on the world has a unique address associated with it. As the length of the world is chosen to be a fixed 10.0, the addresses range from 0.0 to 10.0. As the world is wrapped around, the addresses of its “edge”, namely 0.0 and 10.0 coincide. The Sender (henceforth referred to as ‘S’)

has a pair of “target sensors” and a pair of “agent sensors”. Via one of its target sensors, S receives values $\in [0,1]$ proportional to the absolute value of a “target” address, and on the other target sensor it receives its “complement”, that is, 10 minus the former. Via its agent sensors, S receives values $\in [0,1]$ corresponding to its clockwise and anticlockwise distances to the Receiver (henceforth referred to as ‘R’). R has a similarly functioning pair of agent sensors, but instead of target sensors, it has a pair of “location sensors”. Through them, it receives values $\in [0,1]$ proportional to the absolute value of its own address and its complement, that is, its distance from the edge of the world. The maximum distance through which an agent can sense the proximity of the other (through its agent sensors) is $1/16^{\text{th}}$ of the world length. Each agent has a pair of motors, clockwise and anticlockwise, with which it can cruise around with a maximum possible velocity of $1/64^{\text{th}}$ of the world length at each integration step. Finally, the movement of S is restricted to a special area in the world called the “constraint zone” whose length is a quarter of the world’s length. The zone stretches from the location 1.25 to 8.75 and is thus seated with its center on the edge of the world (see red stripes in fig. 3a and 3b below). In the original work, the constraint-zone was implemented to avoid shepherding-like strategies from evolving. In our setup, since we have a separate communication phase when R is clamped, such strategies can’t evolve. However, since S still has the freedom to move around, a constraint-zone shall contain that as well. Nevertheless, a constraint-zone could be thought of as a model for the bee hive where the dance is performed exclusively.

Each agent has a continuous-time recurrent neural network (CTRNN) based controller with the following state equation (Beer, 1995):

$$\tau_i \dot{s}_i = -s_i + \sum_{j=1}^N w_{ji} \sigma(s_j + \theta_j) + wI_i \quad i = 1, \dots, N$$

where s is the *state* of the neuron, τ is the time constant, w_{ji} is the connection strength between neurons j and i , θ is the bias, $\sigma(x) = 1/(1 + e^{-x})$ is the standard logistic activation function whose value denotes the *output* of the neuron in the range $[0,1]$, and I represents the external input to the neuron scaled with a factor of w . Each agent has 4 sensory neurons (2 agent or target sensors and 2 location sensors), 5 inter-neurons and 2 motor neurons. Of course, the sensory neurons are only input-socket-neurons unlike the rest that are processing-neurons. The inter-neurons are fully inter-connected and each one projects into both the motor neurons. Also, each sensory neuron projects into each of the inter-neurons. In all analyses below, we will use only the outputs of neurons and not their states.

The agent-pairs are evolved using a hill-climbing genetic algorithm with rank-based selection, from a population of 450 agent-pairs. The GA is constrained to search for the CTRNN parameters in the following ranges: connection weights and biases in the range $[-16,16]$, and time constants in $[1,30]$. Mutation variance was varied between 5 and 8 in the searches. Most of the searches were run for about 10000 generations. Each agent-pair is evaluated on a set of 80 “trials”. The trials consist of 8 different pairs of initial S-R locations (within the

constraint zone) and each one headed for 10 different target locations that are equally spaced in the address range $[2.5,7.5]$. Thus, the target range is placed directly opposite to the constraint zone where the communication takes place. At the beginning of each trial, all neuron outputs are set to 0.

Each trial consists of two phases: a “communication phase” and a “courier phase”. The communication phase lasts for 400 time steps (1 through 400) at the beginning of which S and R are placed in such a way that they are always within their proximities but separated by different distances in different trials. Throughout this phase, S constantly receives *fixed* inputs on its target sensors with respect to a particular target address. R receives inputs via its agent sensors only and its location sensors are switched off, during this phase. Further, only S is allowed to move and R is clamped. It is important to note that during this phase, both S and R have no information about their current locations. The following phase, namely the courier phase lasts for 960 time steps (401 through 1360). In this phase, S is removed from the world and the location sensors of R are switched on. R is now allowed to move and is expected to move to the target location whose address was communicated by S during the previous phase. Thus, the fitness score of the GA evaluation is simply an inverse function of the distance between R and the target at the end of the courier phase, averaged over all the 80 trials. Note that the performance of S is not factored in separately as it is implicitly tied to R’s performance. Fig. 1 below shows the evolution of best performance in each run. A few GA searches quickly find good solutions in a few hundred generations but most of them take a couple of thousand generations to find good solutions. Thus, the problem-at-hand, even though designed to look quite simple is somewhat difficult to solve.

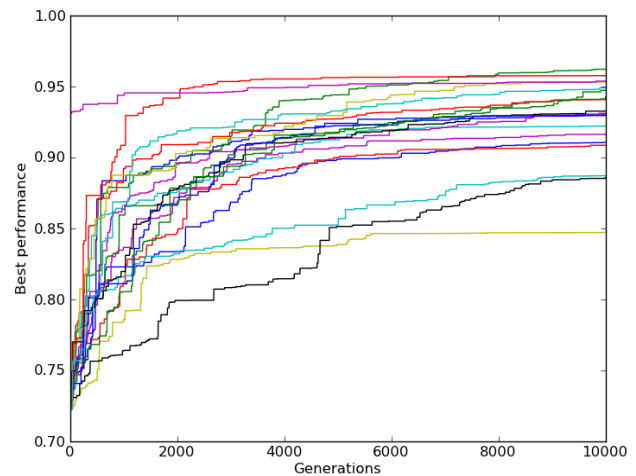


Fig. 1: Evolution results – generations vs. best performance.

Results

The best performing agent-pair achieved a performance of 96% over the 80 trials for which it was evolved. When the agent-pair was evaluated over 300 trials that come from 20 initial S-R separations each heading for a set of 15 targets (within the same range as presented during evolution), its

generalization performance dropped to 93%. Fig. 2 shows the desired vs. actual destinations of R during those trials. It is evident that most of the trips that R makes reach very close to the target, with a few exceptions. Moreover, for every target, there are a few trips that do not land spot-on. Figures 3 show a few sample S-R trajectories. Fig. 3a show four different trials where S communicates four different target addresses to R, from the same initial S-R separation. Fig. 3b show four different trials where S communicates the same target address to R, from four different initial S-R separations.

In all these trials and all others not shown, S likes to move to the right side of R and then move in a characteristic V-shaped trajectory. It is clear from these plots that the width of the V-shape varies with the target address. Fig. 3b also shows the effect of the constraint-zone.

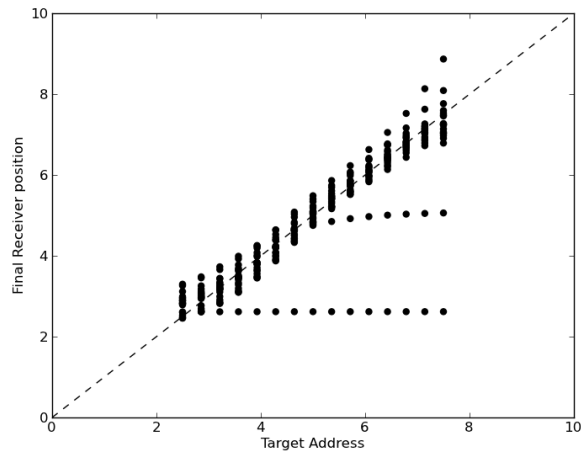


Fig.2: Receiver performance: desired destination vs. actual destination.

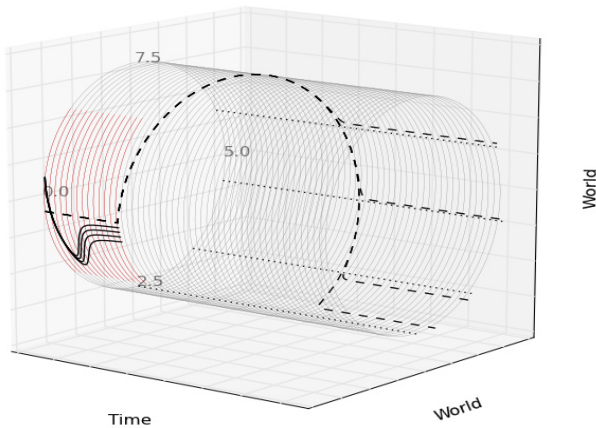


Fig.3a: Sample Sender-Receiver trails for multiple targets. Red stripes represent constraint-zone. Solid lines represent Sender, and dashed lines represent Receiver.

As S starts out more towards the right-edge of the zone, its V-trajectory gets more flattened-out. This has implications on R's behavior and is discussed below. The trajectories of R, as they head towards the targets, do not contain any interesting features except for its consistency in always choosing to move clockwise (towards its left). This behavior is purely an

evolutionary result and no design constraints could have forced it. As S is removed from the world at the end of the communication phase, R does not have the need to make a full round around the circle like what had evolved in the previous work by Williams and Beer. This behavior helps simplify the analysis a bit.

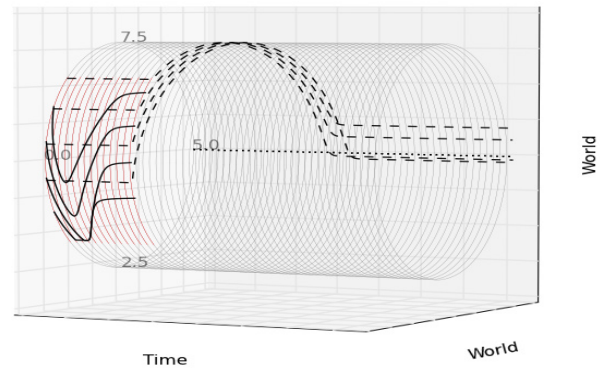


Fig.3b: Sample Sender-Receiver trails for a particular target. Red stripes represent constraint-zone. Solid lines represent Sender, and dashed lines represent Receiver.

Analysis

Receiver behavior

We will focus our analysis almost entirely on the Receiver's behavior, in this paper. Our goal here will be to understand how R stores the information it receives from the Sender and then expresses it in commuting to the target. S' behavior will also be briefly discussed at the end of this section.

Fig. 2 above shows that not all trips of R heading for a particular target land spot-on on the target. The origin of this inaccuracy is rooted deeply in the dynamics of both S and R, which could be exceedingly difficult to nail down. Our goal will be not to show the causes of R's imperfections, but to throw light on how R works when it does best. We will start with certain overt, qualitative aspects of its behavior and then make our first moves into finding what the neurons individually or jointly encode, using information-theoretic tools.

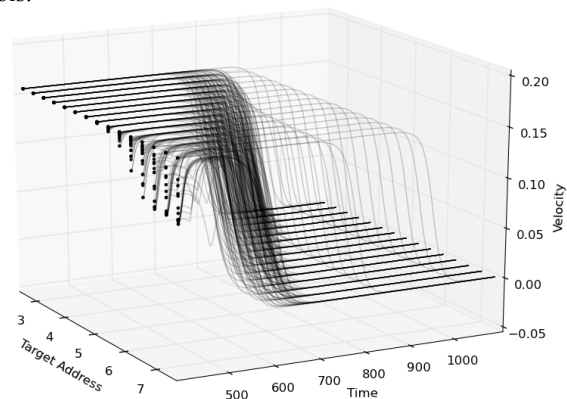


Fig.4: Variation of R's velocity with time, for each target address. The dots denote initial velocities.

As noted earlier, R is able to move during the courier-phase only. This phase starts from $t = 401$ and lasts until $t = 1360$. Fig. 4 above shows how the velocity of R varies with time, as it is heading for each target address from various points in the constraint-zone (see ‘Methods’). As expected, it can be seen that the velocity goes to zero quicker for closer targets (near 7.5) than for the more distant targets. Note that since R always moves in the left direction (see ‘Results’), targets located near 7.5 are considered ‘closer’ and targets located near 2.5 are considered ‘further’ (see ‘Methods’ for addressing-convention). The correspondence of the braking-behavior of R with its initial distance to the targets is better pronounced in fig. 5 below: shorter trips in terms of initial distance halt sooner. Note that an arbitrary distance to target can correspond with multiple targets depending on R’s initial distance to that target. What is also evident in fig. 5 is the correspondence of R’s *initial* velocities (the dots in the plot) with its initial distance to the targets. That is, initial velocity increases with distance-to-target generally (not so for intermediate initial distances ranging from 2 to 4). This makes sense because one would start out slower to commute to closer destinations. However, initial velocities do not smoothly vary with distance for all target addresses. Fig. 4 shows that for a certain set of furthest-away target addresses ranging from 2.5 to about 5.0, the initial velocities are always at a constant maximum.

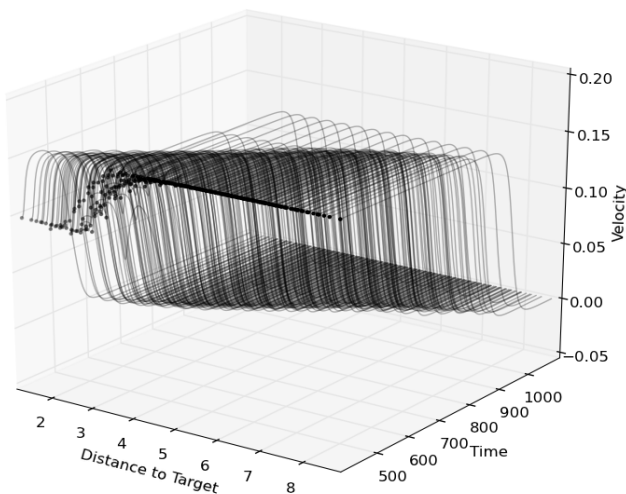


Fig.5: Variation of R’s velocity with time, for every possible **initial** distance to a target address. The dots denote initial velocities.

The above observations suggest that initial velocity depends on both the target address and the initial distance to the target. This is shown in fig. 6 below. The general dependence of initial velocity on the desired distance to travel is reminiscent of a Newton’s equation of motion that relates initial and final velocities, acceleration and distance. In our case, the above figures show that every Receiver achieves the maximum possible velocity (see ‘Methods’) at a visibly constant acceleration; initial velocity then depends on the distance to travel. We still know that distance alone does not determine initial velocity; target plays a role as well. This will be explained information theoretically later in this section. Based on our measurements, we will suggest that the mechanism of

R first discriminates target addresses, then the possible distances to each target.

So far, the story is that S is able to successfully communicate the target address to R by prepping it up with an appropriate initial velocity, which is however not always perfect. At any rate, in order that R goes to the correct destination, it needs to constantly know its current location, as it moves. At the same time, it also needs to constantly remember the target location so it can be compared with the current location. Finally, it needs to make a decision to stop at the right time, based on the result of the comparisons.

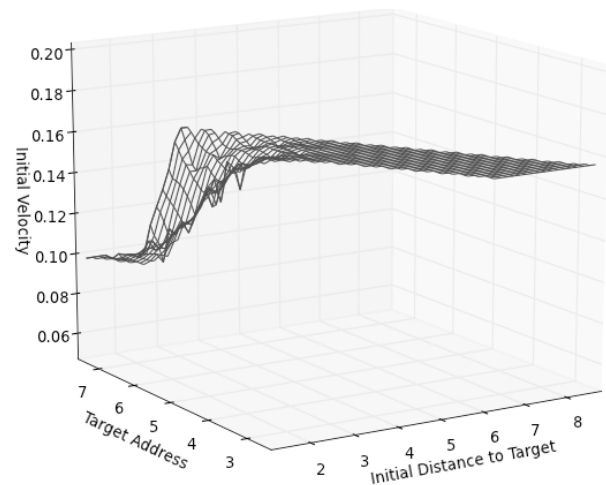


Fig.6: Initial velocity of R is determined by a combination of R’s initial distance to a target and the target address.

Although the velocity of R generally discriminates between target addresses, in fig. 4, it can be seen that there is a small time window between $t = 450$ and 500 when all trips have reached almost the same maximum velocity. This means that the left motor neuron (referred to as ‘ML’ henceforth), which primarily drives R (as it always moves left), produces roughly about the same output during that time window regardless of where R is presently at or where it is heading for. In information-theoretic terms, it can be said that the neuron *ML* doesn’t contain *information* about the target address or current location at least during that time period. However, that information needs to be available at all times. Of the 5 interneurons *N1* through *N5*, one or more of them has to contain that information at any given point in time, either individually or jointly.

We will now define the term “mutual information” that will be used to measure the amount of inter-predictive power between two random variables. Our random variables will hold values of neuron outputs, target addresses or distances to targets. Suppose that X and Y are two random variables that can take any value from their respective ranges and that the values change over time. If at any given time t , we say with some *uncertainty* that X can assume certain values with certain probabilities, then does knowing the value of Y reduce that uncertainty and vice versa? In other words, does knowing the value of X or Y at time t make the other *more* predictable at that instant? A measure of *mutual information* shall help

answer these questions. We say that higher the mutual information between X and Y , higher is this *inter-predictability* between the two, *on an average*. Mathematically, mutual information (MI) is defined as follows (Cover and Thomas, 2006):

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}$$

It can also be defined in terms of uncertainty of a random variable, as follows:

$$I(X;Y) = H(X) - H(X|Y)$$

$H(X)$ is the uncertainty in X , in bits. $H(X|Y)$ is the uncertainty that remains in X when Y is known. X and Y are interchangeable in the above expression. Also, $I(X;Y)$ can never be greater than $H(X)$ because X can at best be fully dependent on or determined by Y , or at worst, is independent of Y . In the best case, $H(X|Y)$ is zero, so $I(X;Y) = H(X)$. In the worst case, $H(X|Y) = H(X)$, so $I(X;Y) = 0$. We will also use an extended form of mutual information called “conditional mutual information”, defined as follows (Cover and Thomas, 2006):

$$I(X;Y|Z) = \sum_{z \in Z} p(z) \sum_{y \in Y} \sum_{x \in X} p(x,y|z) \log_2 \frac{p(x,y|z)}{p(x|z)p(y|z)}$$

Thus $I(X;Y|Z)$ is the *expected* value of $I(X;Y)$ when Z is known. In terms of uncertainty, it is defined as follows:

$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z)$$

That is, it is the amount of uncertainty that Y reduces in X when Z is already known. In fig. 7a below, we show how much mutual information exists between the three different random variables: output of neuron NI (one of the five interneurons), the target addresses, and R 's time-varying distance to the targets. Note that in the plot we show *normalized* mutual information *with respect to* X and $X|Z$, where $I(X;Y)$ is divided by $H(X)$ and $I(X;Y|Z)$ is divided by $H(X|Z)$ to illustrate the proportion of the reduction in uncertainty.

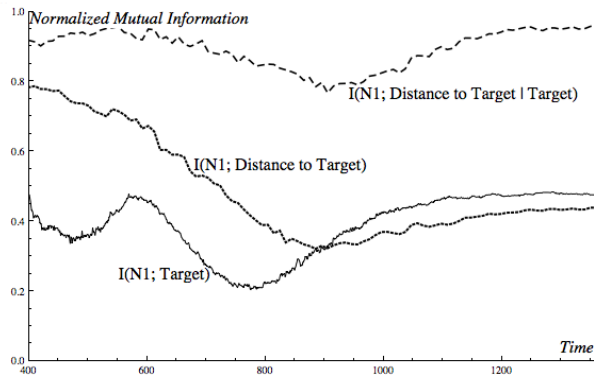


Fig. 7a: Normalized mutual information between NI and (i) Target, (ii) Current distance to target, and (iii) Current distance to target conditioned on target.

Again, note that all measures of MI are computed at *each* time step, where data from all the 300 trials are considered. To estimate the probability distributions, we used quadratic interpolation with the following bin sizes: 0.01 for neuron

outputs, 0.1 for target addresses and 0.05 for distances to targets. We chose these bin sizes based on trial and error until either the actual values of MI between different pairs of variables or the relative differences between them did not change.

As it can be seen, NI contains appreciable amount of information about the target address right from the beginning of the trip. It contains even more information about R 's current distance to its target. These observations should be interpreted as follows: (i) knowledge of target address reduces the amount of uncertainty in the output of NI , given by the value of MI; at $t = 401$, it's about 50%, (ii) knowledge of distance to a target reduces the uncertainty in the output of NI ; at $t = 401$, the reduction is about 80%. Mutual information measures can also be interpreted in terms of sets: each random variable can be represented by a set; MI between two random variables then corresponds to the degree of one-to-one and onto mapping between the two corresponding sets. Since we measure *normalized* $I(X;Y)$ with respect to X , higher MI in our case means that there is a higher degree of one-to-one mapping from X to Y ; we will not be concerned about the onto mapping back from Y to X . Going back to our observations from fig. 7a, we can now say that there is more one-to-one mapping from distance-to-target to NI 's output than from target addresses to NI 's output. Information about the distance to target is crucial because the decision to stop would have to be made in a matter of a few time steps for the closest targets (fig. 5). Since the decision to stop needs to be made sooner for closer targets, it could be fair to hypothesize that information about the current distance to the target depends on the values of the target addresses. That is, $I(NI; \text{Distance-to-Target})$ should depend on target address, and it does, as fig. 7a shows, quite significantly. It should however be noted that the high conditional mutual information also stems from the fact that for a given target address, there are only five different locations (initial distances) that R starts from. Imagine that five Receivers are simultaneously started from those initial locations. Then, based on figures 4 and 5, it can be seen that the inter-distances between the Receivers gradually reduce in a predictable way, as they all move towards the target. That is to say that given a time t , it should be possible to say with much certainty, the exact locations of each of the five Receivers, and in this case that certainty is also reflected in NI 's output. Moreover, even when R comes to a halt, the “residual” distance to target (fig. 2) still determines NI 's output with considerable certainty. This is the reason why the conditional mutual information remains high throughout the trial period. On the other hand, the value of $I(NI; \text{Distance-to-Target})$ that is *not* conditioned on target address drops as time passes. This is probably because, as the clock ticks more and more trips come to a halt resulting in a shrinking of the set (reduction in variety) representing distance-to-target and a corresponding shrink in the degree of uniqueness of the mapping from that set to the NI output set.

We will now briefly discuss a certain relationship between the mechanism of how different values of targets and distances to targets are represented in the Receiver's system and the above information-theoretic measures. It seems possible that evolution has sculpted the mapping of target addresses to NI outputs in such a way that the mapped sub-ranges of NI

output values are somewhat distantly placed. For example, target 2.5 could be represented by $N1$ values in the range $[0,0.1]$, the next target by $[0.1,0.2]$ and so on. The space within each sub-range could then be used to map the various possible distances to the particular target represented by each sub-range. For example, for a target value of 2.5, a distance value of 1.5 could be represented by $[0,0.01]$, the next higher distance by $[0.01,0.02]$ and so on. This is supported by the fact that the MI between $N1$ and distance-to-target is very high when the target address is given (fig. 7a). This also results in a less-than-perfect mapping between distance-to-target and $N1$, because a particular distance-to-target value can correspond with multiple targets (depending on the starting location of the Receiver). That means that given a distance to some arbitrary target, there will be some uncertainty left in determining the value of $N1$'s output. The above hypothesis, if true, could be a reason for why the MI between target address and $N1$ output is not disrupted much throughout the courier phase (fig. 7a). This may be explained as follows. At the end of the communication phase, some target address is conveyed to R, resulting in a $N1$ output in a sub-range corresponding to that target. When the location sensors of R are turned on at $t = 401$, the output of $N1$ is directly affected by the feed from the location sensors. However, the mapping between the new $N1$ output and target addresses does not drastically change as the new values are still within the sub-range corresponding to the target; they only differ according to the distance to that target. These details lead to a hypothesis that is more relevant to the mechanism of R: high MI between target values (that is actually conveyed to R) and neuron outputs of R is not necessary; it can be just sufficient enough and structured appropriately in order there is room for high MI between distance-to-target values, which are more crucial, and the neuron outputs. Thus, the part of the mechanism of R that represents target addresses may act as a *scaffold* to the part that represents distances to targets.

Overall, a few important points of fig. 7a are the following. Neuron $N1$ contains crucial pieces of information necessary for R to function as expected, and is available throughout the trip. With regards to the mechanics of R, an important inference is that it becomes location-aware almost instantly as it begins its trip. This might seem obvious because the "location sensors" of R are connected to all the inter-neurons and therefore one might be tempted to say that the neural activity would have to be determined by the information fed by the sensors. This need not necessarily be the case, as each neuron constantly *integrates* or *accumulates* information from everything it is connected to, and not just the sensors. In our case, R has *evolved* to be highly sensitive to sensory information, as it is crucial. Finally, even though $N1$ loses and gains information, overall it retains it throughout the trip, perhaps with help from other inter-neurons.

In fig. 7b below, we show the same MI measurements that were made for $N1$, for neuron $N5$. Qualitative trends similar to those of $N1$ can be seen for this neuron. This may tempt one to conclude that $N1$ and $N5$ are functionally similar, but it turns out to be not the case, as explained below.

We will now show exactly when the decision to stop is made, and which neurons help it, at least partially. Fig. 8 below

shows the relationship between R's velocity and its current distance to any target. It is evident that most of the trips of R start slowing down when its distance to their targets is about 2.0. Some of them stop short of this distance, and a few overshoot the target (negative distance values). In general, however, there seems to be a consistent pattern to the decision-making. We know that a change in velocity means a change in ML 's output. We also know that at least $N1$ contains the information about the current distance to target, but it does not necessarily mean that $N1$ also makes the stop-decision or equivalently that it directly controls ML 's activity.

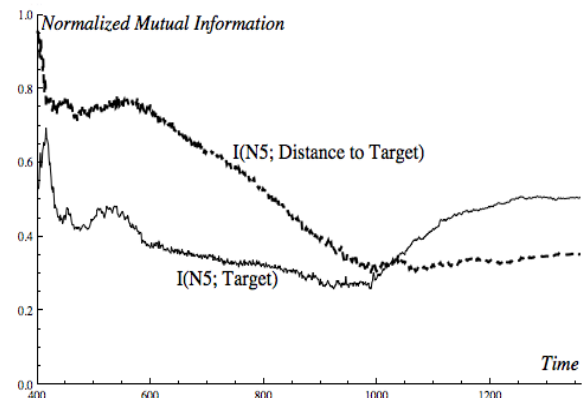


Fig.7b: Normalized mutual information between $N5$ and (i) Target, and (ii) Current distance to target.

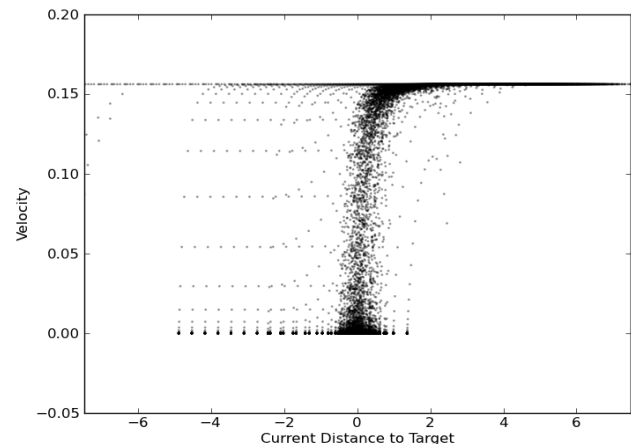


Fig. 8: Current velocity versus current distance to target

However, in our case, $N1$ does take part in the decision-making, but mostly for closer targets. Fig. 9 shows that and also that $N5$ plays a significant role in the decision-making for other distant targets. This is potentially in concert with other neurons too, but we are not analyzing that. In fig.9, we have plotted $I(ML;N)$ when the absolute current distance to target is known and it is a *specific* range of values whose absolute magnitude is ≤ 2.0 . That is, it is a conditional mutual information measure, as described before, but with an added caveat that the conditioning variable takes a particular range of values. This is because we are not interested in understanding how ML is affected when the stop-decision is not about to be made. This approach to measuring conditional

mutual information is a variant of “specific information” used for agent-analysis before (Williams and Beer, 2010).

Earlier we noted that *N1* and *N5* show similar trends in MI between their outputs and the distances to targets. That is, they carry almost the same amount of information about distances, on an average. However, in fig. 9, we can see that when the distance to targets is a particular range of values somewhat close to 0, the information about the distance that *N1* and *N5* carry are employed differently in the system. The difference, as we show, is in the way the information is used to control the neuron *ML*. Neuron *N1* controls *ML* during the earliest instances of the trial and then *N5* also takes over the responsibility during the later instances.

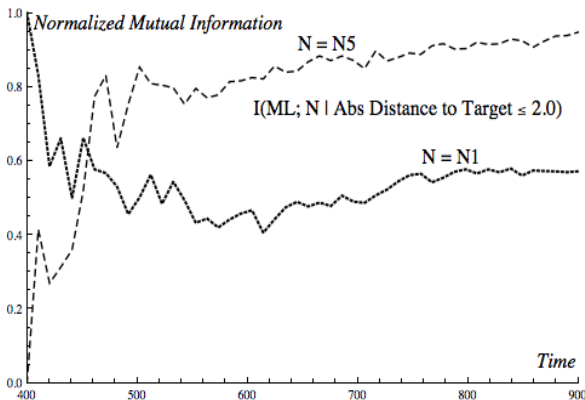


Fig.9: Normalized mutual information between *ML* and *N1/N5* conditioned on absolute distance to target ≤ 2.0

By ‘control’, we mean *causal influence* because there are no physical feedback connections from *ML* to any other neuron; so, a significant MI between an inter-neuron and a motor neuron could be deemed as causal influence. Another important observation is the apparent contradiction between the facts that *N5* carries higher information about distance at $t = 401$ (fig. 7a) than *N1* and yet it has almost no use (zero information about *ML*) unlike *N1*, at the same instant of time. One possible explanation is based on the fact that any measure of MI is only an *average* measure. It is possible that although *N5* contains high distance information on an average, for particular distance values the information is very low. Hence, even though *N1* and *N5* might contain the same amount of information about distances, the actual amounts might vary according to particular values of distances. A final set of observations throw light on the relationship between how inter-neurons control the motor and the overt behavior of *R*. Earlier, we noted that not all the trips of *R* land spot-on on the targets (fig. 2). This is despite the fact that neurons *N1* and *N5* have significant control over the motor throughout the trial. Now, we also know from fig. 7a and 7b that given a distance, some uncertainty remains in determining the outputs of *N1* and *N5*. Hence, it is possible that the uncertainty (variety) in the neuron outputs translate into uncertainty in the motor output (due to high MI with the motor), and therefore in the velocities of *R*, consequently resulting in different landing positions. Further, in fig. 6 we showed that the initial velocity of *R* generally corresponds with distance to target. We know that during the early stages of the trial, *N1* contains high

information about distance and that it also controls *ML*. Thus, the pattern seen in fig. 6 is explained.

Sender behavior

Earlier, we noted that *S* always chooses to move to the right of *R* and perform its “dance” there. Fig. 10 below shows a sample of the not-so-interesting dance. There, we show how for a fixed *S-R* initial separation, *S* moves with respect to *R* (which is also fixed), depending on the target address it is trying to communicate. The green dots in the figure show that *S* is within *R*’s sensory range at those instances. The most conspicuous differences lie in: (i) the time at which *S* re-enters *R*’s sensory range after the small gap during which *S* is not in contact with *R*, and (ii) the distance from *R* at which *S* eventually settles down. Earlier re-entry also means longer time *S* spends with *R* afterward. Since *R* receives most of the communication through its right “agent sensor”, the timing pattern shown above should also be reflected in its activity patterns corresponding to the target address. It should be remembered that the sensory neurons are just socket-neurons, unlike the other inter-neurons, in that they only receive stimulus information from the agent’s exterior and pass it down to the processing-inter-neurons. In fig. 11 below, we show the mutual information between right agent sensor (*SR*) and target address computed from all the 300 trials. As expected, there is a small time window during which MI is zero, and then as time passes, the information about target constantly increases corresponding to the re-entry timing patterns of *S*. That is, as time passes, *R* is able to identify more target addresses that become available.

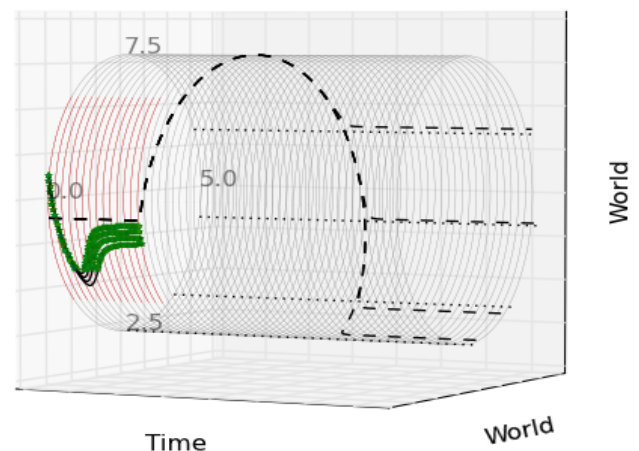


Fig.10: Sample Sender-Receiver trajectories for multiple targets when *S* and *R* start out from the same initial locations. The green ‘stars’ indicate those instances at which *S* is within *R*’s sensory range.

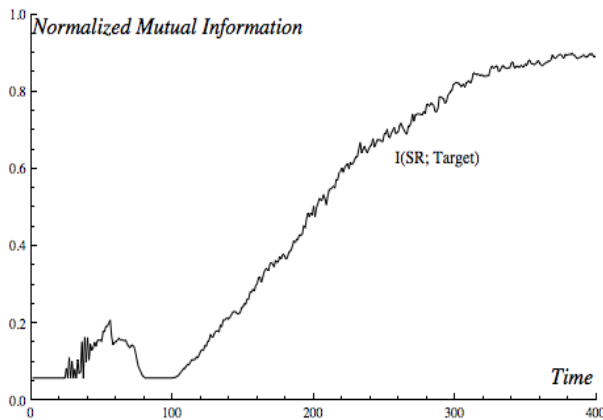


Fig.11: Normalized Mutual information between R's right agent sensor and target address.

Discussion

The problem posed to R is essentially one of number representation (of target addresses) and number comparisons (of distance to target with a threshold). In a programmatic sense, the task is quite simple: it only takes one line of assembly code to implement the logic. However, the agent is not given an operating system that knows how numbers are represented. It only has a system of interconnected parts (neurons) that interact non-linearly. Using this setup, artificial evolution has shaped how S and R represent numbers and how R stores them and uses them to make comparisons. However, it can be said with almost certainty that both S and R will not recognize target addresses beyond the range of numbers that was presented to them during evolution. Nevertheless, even for the small range of numbers that R can understand, it would be fair to hypothesize that R's representation of target addresses would only make sense if it is moving. Suppose that R was clamped even during the courier phase but its location sensors are switched on. What would happen then? Will R's left motor continue to rotate at a constant velocity thinking that it has not reached its target? We would guess otherwise, as R has been evolved to make sense of the target address by moving. Further investigations are needed to answer this question. At any rate, the analyses in the previous section shows that even for this simple task, there could be rich dynamical complexity: information is distributed among neurons, they lose and gain information perhaps by dynamically exchanging them, they could be potentially storing different facets of the same kind of information, they could be assuming different roles over time etc. Of all these possibilities, we have seen only a tiny bit in this work. On the contrary, there is still a possibility that further analyses reveal that certain neurons are redundant, in that they do not play any useful roles or that multiple neurons store the same kind of information. However the neurons dynamically handle information, a successful functioning of the agent demands a corresponding integrity of the agent on the informational front. How that is accomplished amidst the time-varying informational complexity remains to be seen. Moreover, the results of information-theoretic analyses haven't necessarily

thrown light on the mechanical functioning of the system. For example, we know that one of the neurons contains information about the instantaneous distance to the target. However, it is not clear how it is computed: does it involve addition, subtraction? One of the major problems with delving into these kinds of analytical approaches is the combinatorial explosion one would have to face. Even in the simple analysis presented in this work, the candidate neurons for storing target address information were only randomly chosen and analyzed. A thorough analysis would almost certainly show that multiple sets and subsets of neurons engage and disengage "informationally" over time. What could be a principled approach that an investigator should adopt to make such decisions? Is it possible to develop heuristics for such analyses? How would one go about it? Despite these challenges, even preliminary analytical results presented here are encouraging in the sense that there is so much more to understand that are potentially bound to be surprising.

Summary

We started out with the objective of evolving and analyzing Sender-Receiver agent pairs similar to the original work of Williams and Beer (2008). Our successful Sender communicated the target addresses to the Receiver in a characteristic V-shaped trajectory. The Receiver was then able to mostly successfully commute to that location. Information-theoretic analyses of the Receiver showed that the neurons stored information about target address, the time-varying instantaneous distance to the target and they also used that information to make the stopping decision at an appropriate point of time, when the Receiver was at a certain distance away from the target. Depending on whether the target was closer or further, the decision to stop was initiated or caused by different neurons.

Acknowledgements

Thanks to Randy Beer for his intuitive teaching methods and to Paul Williams for providing the source code that was used as a baseline for this work.

References

- Beer, R. D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 3(4):469-509.
- Williams, P. L. and Beer, R. D. (2008). Evolving referential communication in embodied dynamical agents. In S. Bullock, J. Noble, R. Watson and M.A. Bedau, editors, *Artificial Life XI*, pages 702-709. MIT Press, Cambridge, MA.
- Williams, P. L. and Beer, R. D. (2010). Information Dynamics of Evolved Agents. In S. Doncieux, B. Girard, A. Guillot, J. Hallam, J.-A. Meyer and J.-B. Mouret, editors, *SAB'10*, pages 38-49. Springer.
- Cover, T. M. and Thomas, J. A. editors (2006). *Elements of Information Theory*, 2nd edn. Wiley-Interscience, New York.

Effects of Individual Differences on Knowledge and Wisdom of Society: A Social Modeling Approach

Toshihiko Matsuka and Hidehito Honda

Department of Cognitive and Information Science
Chiba University, Chiba 261-8522, JAPAN
{matsuka, hito}@muscat.L.chiba-u.ac.jp

Abstract

Categorically organized knowledge is the main vehicle in high-level cognitive processes. The previous empirical and theoretical studies on categorization paid almost exclusive attention to how individuals learn categorical knowledge. In the real world, however, people acquire knowledge not only through individual learning, but also through interacting with others. In the present study, using computational modeling, we explored how social interactions would produce unique dynamics of knowledge acquisition that cannot be examined by studies on micro level processes. The results of simulation studies showed that when there were several clusters of individuals in a society where individuals held different beliefs about what constitutes "good" knowledge, then the society as a whole formed Pareto-optimal knowledge. That is, there was no cluster of knowledge that was simultaneously worse in two important aspects of knowledge (i.e., accuracy and simplicity) as compared with those of other clusters in a mature society.

Introduction

Categorically organized knowledge is the main vehicle in high-level cognitive processes, such as reasoning and communication (e.g. Murphy, 2002). Categorical knowledge which is often referred to as *concepts* allow us to achieve very complex cognitive tasks by effectively compressing overwhelmingly abundant information into manageable and meaningful chunks. Because of its importance, cognitive processes associated with categorization have been widely studied in the area of Cognitive Science, both empirically with behavioral experiments and theoretically with computational modeling techniques. In the empirical studies, researchers usually create experimental settings where individual participants learn categories by corrective feedback, providing empirical evidence about how people acquire knowledge through individual learning (e.g., Cohen & Lefebvre, 2005). Using the results of these empirical studies, computational studies also pay almost exclusive attention to how individuals learn categorical knowledge.

However, in the real world, people acquire categorical knowledge not only through individual learning, but also through interacting with others. Pentland (2007) argued

that influences of social structures and activities need to be considered in order to better understand true human cognitive behaviors. Likewise, Goldstone and Janssen (2005) emphasized the importance of research on collective behavior. For example, they pointed out that "interacting ants create colony architectures that no single ant intends," indicating that social interactions can produce unique dynamics of knowledge acquisition that cannot be clarified by studies on individual's micro-level processes in knowledge acquisition.

In the present paper, we examine how a society as a whole would acquire categorical knowledge where some degree of individual differences exist in the society.

Computational Models

In the present paper, we used ALCOVE (Kruschke, 1992) as the model of individuals' categorization processes, and an optimization method based on evolutionary computation techniques as the model of social learning processes.

Individuals' Categorization Algorithm - ALCOVE

ALCOVE is a computational model of category learning that assumes that humans store many previously seen or experienced exemplars in their memory, and categorize input stimuli on the basis of psychological similarities between the inputs and the memorized exemplars. Psychological distances between an input stimulus and those memorized exemplars activate exemplar nodes in ALCOVE. Exemplars that are "psychologically" similar to an input are more highly activated than exemplars that are "psychologically" dissimilar. Specifically, as shown in Eq. 1, j th exemplar's activation (h_j) in ALCOVE is based on the inverse distance between an input, x , and a stored exemplar, ψ_j , in multi-dimensional representational space where each dimension (i) is scaled by non-negative selective attention weights, a_i :

$$h_j^{(m)}(x) = \exp \left(-\beta \cdot \sum_{i=1}^I a_i^{(m)} |\psi_{ji} - x_i| \right) \quad (1)$$

where β is called specificity which determines an overall similarity gradient, and superscript m indicates a categorization strategy or knowledge held by a particular individual

m . Because our learning algorithm is built on the basis of a stochastic optimization technique, dimensional attention weights take the following form to attain stability in the model's behaviors:

$$a_i^{(m)} = \left(1 + \exp\left(-D_i^{(m)}\right)\right)^{-1} \quad (2)$$

where D_i is a pseudo-attention weight that is being updated in learning (not as).

The exemplar activations are then fed forward to the k -th output node (e.g., output for category k), O_k , weighted by w_{kj} , which determines the strength of association between exemplar j and output node k :

$$O_k^{(m)}(x) = \sum_{j=1}^J w_{kj}^{(m)} h_j^{(m)}(x) \quad (3)$$

The probability of categorizing input instance x to category C is based on the activation of output node C relative to the activations of all output nodes:

$$P(C|x) = \frac{\exp\left(\phi \cdot O_C^{(m)}(x)\right)}{\sum_k \exp\left(\phi \cdot O_k^{(m)}(x)\right)}. \quad (4)$$

where ϕ controls decisiveness of the classification response. Higher ϕ values cause more extreme decisions.

Learning Algorithms

Overview of Learning Algorithm In the present research we assumed that quite simple learning processes take place in a society. In particular, we assumed that people communicate and exchange their knowledge with others where each individual would combine his or her knowledge with that of another individual. We refer to this process as "Knowledge Combination." After combining knowledge, each individual is assumed to modify his or her own knowledge by randomly altering it. We refer to this process as "Knowledge Modification." Knowledge Combination and Modification together may be interpreted as formations of new hypotheses. Finally, we also assumed that each individual has their own belief about what constitutes "good" knowledge, and knowledge that is believe to be good will be kept by individuals and therefore by the society. We refer to this process as "Knowledge Selection."

In modeling the abovementioned learning strategies that take place within a society, we incorporated a type of Evolution Strategy (ES) techniques in the present research. An ES is a type of evolutionary computation method that is typically used for continuous parameter optimization. Knowledge Combination is achieved by what is called *crossover* in evolutionary computation literature in which randomly selected two individuals exchange their knowledge (i.e., parameters or coefficients in ES). Knowledge Modification is

achieved by a process called *mutation* in which a small random value drawn from the Normal distribution is added to each element of knowledge (i.e., parameter). After new knowledge is formed through Knowledge Combination and Modification, each individual assesses his or her own knowledge on the basis of self-defined knowledge utility. Knowledge with high utility values will be kept by individuals and the society, while that with low utility values will be discarded.

Social structure There are few assumptions about how a society is organized. We assumed that people have interactions with a limited number of individuals, forming clusters of individuals. In other words, our model of a society has a highly locally clustered structure like a small world network (Watts & Strogatz, 1998). Previous studies have shown that many real world networks have analogous network structure to a small world network. For example, collaboration networks of film actors (Watts & Strogatz, 1998), networks of scientific collaboration (Newman, 2001), and ownership links among German firms (Kogut & Gordon, 2001) are shown to be structured as small world networks.

We further assumed that the principle of homophily exists in a society such that people who have similar beliefs (about constitutes "good" knowledge) would have close relationships with each other and that those who have close relationships would learn from each other. This assumption has reasonable face validity as, for example, right-wing conservatives often omit what is being stated by left-wing liberals or vice versa. For the sake of simplicity we assumed that people exchange information only with people from the same cluster, meaning that there are several independent or segregated clusters in a society (thus, although there several local clusters within a society like a small world network, our model of a society is not organized as a small world network as individuals from different clusters are not connected). People within the same cluster have the similar beliefs about constitutes good knowledge, while different clusters of individuals possess different beliefs. Knowledge Combination and Knowledge Selection take place within clusters (Knowledge Modification takes place within individuals).

Knowledge Combinations In Knowledge Combination, randomly selected pairs of individuals within a cluster exchange information to form new knowledge. For the sake of simplicity, we use the following notation $\{\mathbf{w}^{(m)}, \mathbf{D}^{(m)}\} \in \boldsymbol{\theta}^{(m)}$. The model utilizes discrete recombination for knowledge parameters (i.e., θ s). Thus,

$$\theta_l^{(c)} = \begin{cases} \theta_l^{(p1)} & \text{if UNI} \leq 0.5 \\ \theta_l^{(p2)} & \text{otherwise} \end{cases} \quad (5)$$

where UNI is a random number drawn from the Uniform distribution. For self-adapting strategy parameters (i.e., σ s), intermediary recombination (simple arithmetic average) is

used, thus $\sigma_i^{(c)} = 0.5 \cdot (\sigma_i^{(p1)} + \sigma_i^{(p2)})$. The parameters for self-adaptation are the parameters that define search widths (i.e., learning rates) for the parameters for knowledge (i.e., \mathbf{w}, \mathbf{D}). A unique search width is allocated to each association and attention weight within individuals so that sensitivity to objective hypersurface is individually tailored to meet his or her learning objectives.

This combination process continues until the number of new knowledge produced reaches the memory capacity of the model.

Knowledge Modifications After Knowledge Combination, each individual randomly modifies his or her knowledge, using a self-adapting strategy. Thus,

$$\sigma_{\theta_l}^{(m)}(t+1) = \sigma_{\theta_l}^{(m)}(t) \cdot \exp(N(0, \gamma)) \quad (6)$$

$$\theta_l^{(m)}(t+1) = \theta_l^{(m)}(t) + N(0, \sigma_{\theta_l}^{(m)}(t+1)) \quad (7)$$

where t indicates time, l indicates parameters, γ defines search width (via σ 's), and $N(0, \sigma)$ is a random number drawn from the Normal distribution with the corresponding parameters.

Knowledge Selection

We assumed that there are two "universally" important elements in determining utility of knowledge on categorization. One is accuracy and the other is simplicity. Everyone, regardless of his or her belief about what constitutes good knowledge, evaluates his or her knowledge on the basis of those two elements. However, individuals from different clusters differently weight the importance of those two elements. In the present research we operationally define different beliefs by different sets of weight vectors

Accuracy (inaccuracy) In the model, inaccuracy (thus accuracy) of a particular set of parameters (knowledge) is estimated based on a set of all unique exemplars in a training set (i.e., errors in batch learning). Thus, knowledge inaccuracy is given as follows:

$$E(\theta^{(m)}) = \sum_{n=1}^N \sum_{k=1}^K \left[d_k^{(n)} - P(k|x^{(n)}) \right]^2 \quad (8)$$

where superscript n indicates a particular input-output pair, N is the number of unique training pairs, and d_k is the desired output value ('1' if for k is a correct category, and 0 otherwise) for category k , and $P(k|x^{(n)})$ is a probability that input $x^{(n)}$ being categorized as k . The desired output values are assumed to be obtained individually and thus Knowledge Inaccuracy is individually estimated.

For modeling individual learning processes, a batch learning method may be not psychologically valid (e.g., Matsuka, Sakamoto, Chouchourelou, & Nickerson 2008). In order to more precisely model individual's learning processes, this

inaccuracy function can be easily extended to include a retrospective verification process (e.g., Matsuka, et al., 2008) that simultaneously accounts for laws of learning and forgetting (Anderson & Schooler, 1991).

Simplicity (complexity) There are two separate elements that defines knowledge simplicity (complexity), one based on association weights and other based on attention weights. The complexity measure based on association weights is as follows:

$$Comp_w^{(m)} = \sum_{kj} \left(w_{kj}^{(m)} \right)^2 \quad (9)$$

This complexity measure simply signify absolute magnitudes of association weights. Thus, when exemplar nodes and category nodes are weakly associated in general, this measure tends to be small. This measure does not directly take into account the number of exemplars memorized and utilized. On the other hands, the complexity measure for attention weights take into account the number of feature dimensions being attended.

$$Comp_a^{(m)} = \sum_i \frac{\left(a_i^{(m)} \right)^2}{\left(a_i^{(m)} \right)^2 + \sum_{l=1}^I \left(a_l^{(m)} \right)^2} \quad (10)$$

This measure tends to be small when a smaller number of feature dimensions is selectively attended. Note that this measure is estimated based on selective attention weights a s, but not pseudo-selective attention weights D s.

The overall knowledge complexity is the sum of two complexity measures, thus $Comp(x^{(m)}) = Comp_w^{(m)} + Comp_a^{(m)}$.

Individual Differences in Learning Objectives Although we assumed that all individuals take both accuracy and simplicity into account in learning, there are some individual differences in weighting those two elements. We consider the difference in weights corresponds to difference in their beliefs. We define v_E^κ as a scaler weighting for relative importance of Knowledge Inaccuracy, and $v_{comp}^\kappa = 1 - v_E^\kappa$ for Knowledge Complexity.

Using these weights and Knowledge Inaccuracy and Complexity measures, we let

$$F(x^{(m)}) = v_E^\kappa E(x^{(m)}) + v_{comp}^\kappa Comp(x^{(m)}) \quad (11)$$

as an overall fitness value of knowledge for a given belief (a particular Inaccuracy - Complexity weighting vector).

Simulation

In order to explore how social interactions would produce unique dynamics of knowledge acquisition, two simulation studies were conducted. In both simulation studies, the

Table 1: Schematic representation of stimulus set used in Simulation 1

Cat	D1	D2	D3	D4
A	1	1	1	0
A	1	0	1	0
A	1	0	1	1
A	1	1	0	1
A	0	1	1	1
B	1	1	0	0
B	0	1	1	0
B	0	0	0	1
B	0	0	0	0

model, thus, a society was given simple categories to learn. In Simulation 1, we examined characteristics of knowledge acquired by the society as a whole, using a stimulus set from a classical study (Medin & Schaffer, 1978). Simulations 2 was conducted to confirm the results of Simulation 1 and to propose a new way to analyze the properties of category structures.

Simulation 1

Method Table 1 shows schematic representation of stimulus set, which was adapted from Medin & Schaffer (1978). The model was run in a simulated training procedure with 500 trial blocks (generations), where each block consisted of a random presentation of the nine unique training exemplars (see Table 1) exactly once, in order to learn the categories. The model parameters were arbitrary selected: $\beta = 2.0$, $\phi = 3.0$, $\gamma = 0.1$. There were 50 clusters within which there were 10 simulated individuals, thus there were a total of 500 individuals in Simulation 1. The scaler weights that define relative importance for Knowledge Accuracy (i.e., v_E^κ) were evenly spread from 0 and 1 for the 50 clusters (i.e., 0.000, 0.0204, 0.0408, ..., 1). Note that the weight for Knowledge Complexity was 1 minus Knowledge Accuracy ($v_{comp}^\kappa = 1 - v_E^\kappa$).

Results and Discussion Figure 1 shows characteristics of knowledge acquired by individuals in a society, where each dot represents knowledge acquired by one individual and knowledge characteristic of every individual is plotted. The vertical axis represents error (i.e., Knowledge Inaccuracy), while the horizontal axis represents Knowledge Complexity. The figure shows that there was a great degree of individual differences in acquired knowledge. Some individuals acquired very accurate knowledge at the cost of complexity, while others acquired very simple knowledge at the cost of accuracy. It also shows that the society as a whole formed Pareto-optimal knowledge. That is no individual acquired knowledge that was worse in both Knowledge Inaccuracy and Knowledge Complexity as compared with those of other individuals, or no individual acquired knowledge

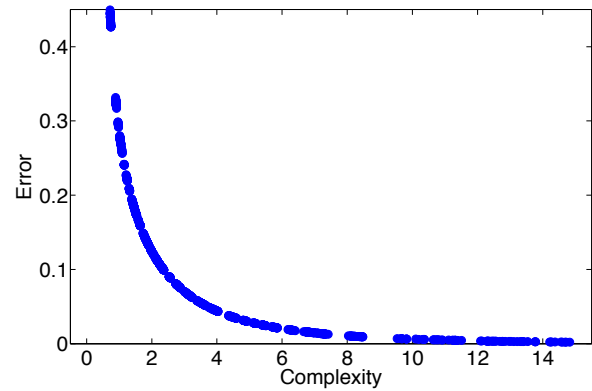


Figure 1: Results of Simulation 1. This figure shows characteristics of knowledge acquired by a society, where each dot represents knowledge acquired by one individual. Some individuals acquired very accurate knowledge at the cost of complexity, while others acquired very simple knowledge at the cost of accuracy. The results shows that the society as a whole formed Pareto-optimal knowledge.

that was better in both Knowledge Accuracy and Knowledge Simplicity as compared with others. The results can be interpreted as that a society would acquire cluster of knowledge that exceed at least one important aspect of knowledge when there are individual differences in beliefs and values and when individuals learn from others who share similar beliefs and values. This result was not surprising, because our model resembles one of multi-objective evolutionary optimization methods called vector evaluated approach (Deb, 2001). The resemblance may indicate that the principle of homophily (i.e., people who have similar beliefs tend to have close relationships with each other) and individual differences together can lead a society to acquire and hold pareto-optimal knowledge.

Another interesting result was that there were some individuals who did not have any clue about categories (i.e., individuals whose knowledge accuracies were at the chance level). Although it may sound a bit odd that some individuals did not learn this type of simple categories, the result is very much expected because those individual did not care about how accurate their knowledge was (i.e., $v_E^\kappa = 0$) as long as their knowledge was at a minimum complexity ($v_{comp}^\kappa = 1$). This type of individuals is uncommon in a society - some people are ignorant about certain things. Using a social simulation approach, we were able to reproduce a wide variety of individuals with different types of knowledge about categories.

Simulation 2

Simulations 2 serves two purposes. One is to confirm the results of Simulation 1. The other is to propose a new way to

Table 2: Schematic representation of stimulus set used in Simulation 2

Stimulus Features			Category Types					
Dim1	Dim2	Dim3	T1	T2	T3	T4	T5	T6
1	1	1	A	A	A	A	A	A
1	1	2	A	A	A	A	A	B
1	2	1	A	B	A	A	A	B
1	2	2	A	B	B	B	B	A
2	1	1	B	B	B	A	B	B
2	1	2	B	B	A	B	B	A
2	2	1	B	A	B	B	B	A
2	2	2	B	A	B	B	A	B

analyze the properties of category structure and/or cognitive demands required by learning categories.

Method Table 2 shows schematic representation of stimulus set, which was adapted from Shepard Hovland & Jenkins (1961). These six categories differ their complexities for correct categorizations. The results of previous empirical studies showed (Nosofsky, Gluck, Palmeri, McKinley, Glauthier, 1994; Shepard et al., 1961) that Type 1 (T1) was the easiest to learn to classify, followed by T2, T3, T4, T5, and T6 being the most difficult, where the differences in difficulties for T3, T4, and T5 were not statistically significant. Because T3, T4, and T5 were not statistically significant, we used only T3 among those three categories in Simulation 2. T1 was easiest to learn, probably because it only requires a simple one-dimension rule for a correct categorization (i.e., an input stimulus is "Category A," if Dim is "1," or "Category B," if "2"). T2 can be considered as XOR-logic, described by Dimensions 1 and 2. T3 is one-dimensional rules with two exceptions (one for each category) where recognition of the exceptions requires consideration of all three feature dimensions. T6 was the most complex as it requires memorization of many if not all exemplars.

The model was run in a simulated training procedure with 500 trial blocks, where each block consisted of a random presentation of the eight unique training exemplars (see Table 2) exactly once, in order to learn the category. The parameter values in Simulation 2 were identical to Simulation 1. There were 50 clusters within which there were 10 simulated individuals, thus there were a total of 500 people in Simulation 2. The weights that define relative importance for Knowledge Accuracy were evenly spread from 0 and 1 for the 50 clusters.

Results and Discussion

Figure 2 shows characteristics of knowledge acquired by individuals in a society, where each dot represents knowledge acquired by one individual. As in Simulation 1, some individuals acquired very accurate knowledge at the cost of

complexity, while others acquired very simple knowledge at the cost of accuracy, resulting in Pareto-optimal knowledge acquisition by a society in Simulation 2. This confirms that the principle of homophily and individual differences together can lead to acquisition of pareto-optimal knowledge by a society.

Four separate pareto-front lines were resulted from learning four categories. Given that the simulated learning processes were minimization problems (minimizing Knowledge Inaccuracy and Complexity), a line that is closer toward 0s in both objectives represent a category that is easier to learn, where easiness is defined by complexity relative to inaccuracy or vice versa. Thus, Simulation 2 replicated the order of difficulties for those categories suggested by empirical (Nosofsky et al., 1994; Shepard et al., 1961) and theoretical studies (e.g. Feldman, 2003). This implies that our simulation method can be used as a tool to analyze characteristics of category structures and/or to evaluate psychological validities of models of categorization or category learning. In fact, when a typical prototype model, which assumes that people hold one prototype for each category and categorize an input on the basis of psychological similarities between the input and the prototypes, was used for simulations, T3 was found to be "easier" to learn than T2, being inconsistent with empirical findings. This result suggests that a typical prototype model of categorization is inadequate in describe real human cognitive behaviors.

What is prominent our approach is that, unlike traditional theoretical approaches that are built on the basis of normative accounts (i.e., how human *should* think or behave), it can incorporate even subjective beliefs and attitudes into objectives of learning as long as they are consistent with real human cognitive behaviors. In other words, a social category learning simulation paradigm that incorporate the principle of homophily and individual differences is an effective exploratory tool in examining the nature of our bounded cognitive rationality and cognitive demands required by realistic contexts and situations.

Conclusion and Future Directions

Categorically organized knowledge is inarguably the main vehicle in high-level cognition. Unlike previous studies which primarily focus on individual learning processes, we examined learning processes that take place in a society. In so doing we assumed that the principle of homophily (i.e., people who have similar beliefs tend to have close relationships with each other) and individual differences exist in a society. In two simulation studies that incorporated those two characteristics, we found that the society would acquire pareto-optimal knowledge, such that no cluster of knowledge that was worse in two important aspects of knowledge (i.e., accuracy and simplicity) as compared with those of other clusters.

In addition, our social category learning simulation was

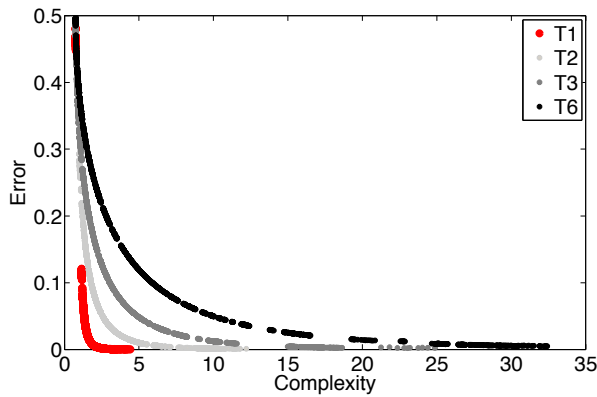


Figure 2: Results of Simulation 2. As in Simulation 1, some individuals acquired very accurate knowledge at the cost of complexity, while others acquired very simple knowledge at the cost of accuracy. There were four separate pareto-front lines for four types of categories. It replicated the order of difficulties suggested by empirical and theoretical studies.

found to be an effective exploratory tool in examining the nature of our bounded cognitive rationality and cognitive demands required by realistic contexts and situations.

A natural extension of the present study is to examine other types of social structure, including small world networks (Watts & Strogatz, 1998) and scale free networks (Barabasi, & Albert, 1999). The principle of homophily and individual differences are not uncommon in a society, but presence of clearly segregated clusters might not have been realistic. Honda and Matsuka (2011) showed that when a network consists of several clusters (i.e., a small world network) a society as a whole can maintain diverse knowledge. Although, their simulation studies paid more attention to structure of networks and incorporated rather simple learning algorithms, we expect somewhat similar findings when we use small world networks in our simulation paradigm. Additional simulation studies are needed to confirm this speculation and to see the dynamics of knowledge acquisition in a scale free network.

In the present study, we showed that presence of the principle of homophily and individual difference are robust characteristics of a society that leads to acquisition of pareto-optimal knowledge.

Acknowledgements

This work was in part supported by the Japan Society for the Promotion of Science KAKENHI (Grant No. 23500335) and Nakajima Foundation.

References

Anderson, J. R. & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, 2, 396-

408.

- Barabasi, A. & Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286, 509–512.
- Cohen, H., & Lefebvre, C. (Eds.). (2005). *Handbook of Categorization in Cognitive Science*. Elsevier Science.
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*, Chichester: Wiley.
- Feldman, J. (2003). The simplicity principle in human concept learning. *Current Directions in Psychological Science*, 12 227-232.
- Goldstone, R. L., & Janssen, M. A. (2005). Computational models of collective behavior. *Trends in Cognitive Sciences*, 9, 424-430.
- Honda, H. & Matsuka, T. (2011). How is knowledge transmitted in a small world network through communicative interaction? In the *Proceedings of the 33rd Annual Conference of the Cognitive Science Society* (pp. 1841 - 1846). Austin, TX: Cognitive Science Society.
- Kruschke, J. K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review*, 99, 22-44.
- Kogut, B., & Gordon, W. (2001). The small world of Germany and the durability of national networks. *American Sociological Review*, 66, 317-335.
- Matsuka, T., Sakamoto, Y., Chouchourelou, A., & Nickerson, J. V. (2008). Toward a descriptive cognitive model of human learning. *Neurocomputing*, 71, 2446-2455.
- Medin, D. L., & Schaffer, M. M. (1978). Context Theory of Classification Learning. *Psychological Review* 85, 207-238.
- Murphy, G. L. (2002). *Big book on concept*. Cambridge, MA: MIT.
- Newman, M. E. J. (2001). The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98, 404-409.
- Nosofsky, R. M. Gluck, M. A., Palmeri, T. J., McKinley, S. C., & Glauthier, P. (1994). Comparing models of rule-based classification learning: A replication and extension of Shepard, Hovland, and Jenkins (1961), *Memory and Cognition*, 22 352-36.
- Pentland, A. (2007). On the Collective Nature of Human Intelligence. *Adaptive Behavior*, 15, 189-198.
- Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393, 440-442.

With a little help from selection pressures: evolution of memory in robot controllers

Ollion Charles^{1,2}, Pinville Tony^{1,2} and Doncieux Stéphane^{1,2}

¹- UPMC Univ Paris 06, UMR 7222, ISIR, F-75005, Paris, France

²- CNRS, UMR 7222, ISIR, F-75005, Paris, France
ollion,pinville,doncieux@isir.upmc.fr

Abstract

Evolutionary robotics (ER) have successfully built robot controllers presenting a reactive behavior. However, the evolution of cognitive controllers is still a challenge. We hypothesize here that a fitness function which rewards the fulfillment of a task requiring cognitive abilities does not necessarily reward the stepping stones that lead to cognitive controllers. In other words, our hypothesis is that evolving cognitive abilities is a deceptive problem, and that the selective pressures driving the evolutionary search are of critical importance. This paper presents some experiments to confirm this hypothesis and addresses this selective pressure problem by introducing a new helper-objective that rewards controllers with a memory. This is potentially useful for the design of controllers in which an internal representation of some data is required to solve a task. It does not assume how the memory is stored in the controller, therefore reducing the bias towards a particular solution. The new objective is tested in a multi-objective scheme on a T-maze ER task — a task involving both navigation and working memory. The efficiency of the helper-objective is studied, as well as its effects on the overall performance and generalization ability of the controller.

Introduction

Evolutionary Robotics deals with the use of evolutionary algorithms (EA) in the design process of robots (Doncieux et al., 2011; Floreano et al., 2008b). Such algorithms have been used for various tasks (Nelson et al., 2009). Typical studies deal with the evolution of locomotion controllers (Allen and Faloutsos, 2009) or obstacle avoidance controllers, (Durand et al., 2000), that are mostly reactive behaviors, i.e. behaviors that usually do not require any memory of past actions or perceptions.

Cognition may refer to a wide range of abilities: from capacities specific to humans (Dennett, 1997) to any ability of a living organism (Maturana and Varela, 1980; Heschl, 1990). Here we will use it to describe abilities that go beyond reactive behaviors and require to take past actions and/or perceptions into account while choosing how to move and what to do. A neural network can exhibit such abilities thanks to a recurrent network structure or to some form of plasticity. The question we will address here is the evolu-

tion of such cognitive abilities with a focus on non-plastic recurrent networks.

Following the seminal work of Yamauchi and Beer (Yamauchi and Beer, 1994), most works on this topic have focused on network structures (Ziemke, 1999; Capi and Doya, 2005a). One problem remains when evolving such systems: the evolvability. Evolving such networks is a challenge (Blynell and Floreano, 2003) and we may wonder why. Evolutionary search proceeds by balancing diversification, which consists of exploring the search space, with intensification, which consists of optimizing the best solutions found so far. These two different aspects of EA result from the exploration done by the genetic operators (mutation and cross-over) together with the selection algorithm that relies on fitness values. We will refer to the fitness function and all mechanisms influencing the selection process as *selection pressures*. In this work, we will hypothesize that the difficulty of generating cognitive abilities is not (at least not only) a problem of network structure or encoding, but rather a problem of selection pressure. The question we will address is then: *what selection pressure to use to drive the evolutionary search towards controllers with cognitive abilities?*

A selection pressure should drive the evolutionary search from randomly generated individuals to desired solutions. We hypothesize here that evolving cognitive abilities is a deceptive task, i.e. that intuitive goal oriented fitness functions are misleading. More precisely, we think that reactive controllers represent a very attractive local optima that is difficult to escape from and the contribution of this work aims at enhancing both diversification and intensification phases to solve this problem. The first contribution consists in showing the impact of behavioral diversity (Mouret and Doncieux, 2012) for the evolution of cognitive abilities, while it has been tested mostly on reactive controllers up to now. Behavioral diversity is a selection pressure that is independent from the cognitive abilities we are looking for. The second contribution is the proposition of a new selection pressure dedicated to the emergence of an internal representation. This selection pressure explicitly rewards networks

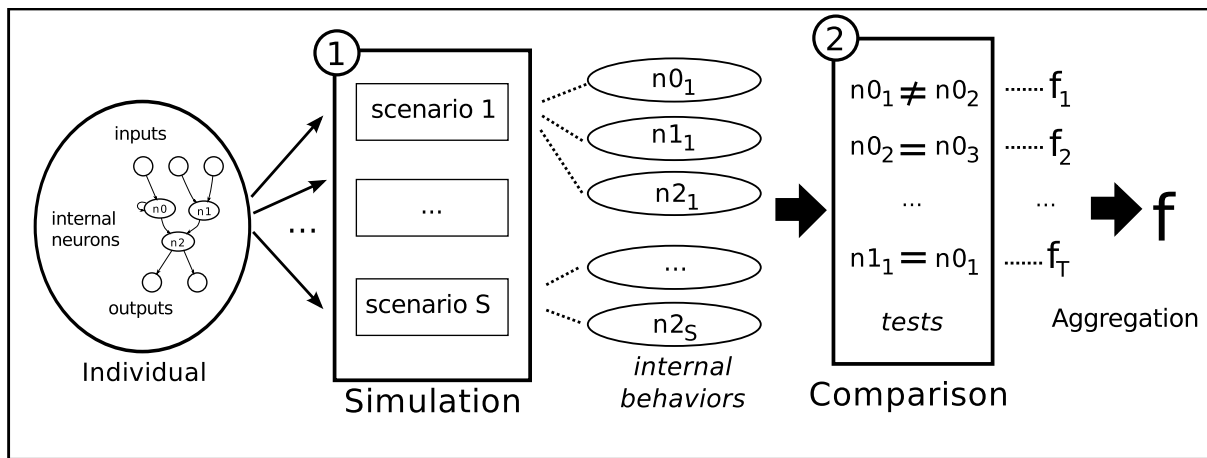


Figure 1: Details of the evaluation of an individual by the scenario-based objective. 1) An individual (here a neural network with internal neurons n_1, n_2, \dots) is simulated in several predefined scenarios. During this simulation, the behavior of each internal neuron is stored. 2) The internal behaviors are compared and checked for coherence, resulting in a partial fitness value f_i . Then, the partial fitness values are aggregated into the final evaluation f .

that exhibit some form of memory. It has been designed with the goal to be compatible with any kind of neural network encoding and without making any a priori on where the memory should emerge. These two contributions have been tested on a T-maze navigation task requiring to memorize some inputs to generate the expected behavior.

Related Work

Evolution of cognitive abilities

Two main kinds of cognitive abilities have been investigated so far: memory (Ziemke, 1999; Ziemke and Thieme, 2002) and learning (Blynel and Floreano, 2003; Floreano and Urzelai, 2001). These works proved that it was possible to generate such capabilities with an evolutionary approach, but it still remains a challenge. Such contributions can be roughly divided in two different categories (Floreano et al., 2008a). Following the seminal work of Yamauchi and Beer (1994), the first category studies continuous time recurrent neural networks without plasticity. Multiple experiments have thus shown that, through evolutionary optimization, such networks can exhibit a memory capability (Ziemke, 1999; Blynel and Floreano, 2003; Capi and Doya, 2005a). The second category focuses on learning and neuromodulation and tries to evolve networks with plastic connections (Floreano and Urzelai, 2000; Ziemke and Thieme, 2002; Tonelli and Mouret, 2011). Both kinds of work mainly focus on the features of the neural network structure (completely connected neural network, Elman network or others) or on the encodings that allows to explore network structures with evolutionary algorithms.

Selection pressures

Floreano and Urzelai (2000) proposed a framework for describing fitness functions: *the fitness space*. Recognizing the importance of the fitness function definition on the results of an ER experiment, they proposed a classification of fitness functions in order to easily allow their qualitative description, assessment and comparison. Nelson et al. (2009) have made a review of the different fitness functions used in ER classified according to the degree of a priori knowledge incorporated in the fitness. Both works recognized the impact on performance of the fitness function, but none of them aimed at better understanding it.

Lehman and Stanley (2008, 2011) have shown how deceptive goal-oriented fitness functions can be. The novelty search approach they have proposed consists in looking for novel solutions instead of efficient ones. Associated with the increasing complexity feature of the NEAT encoding (Stanley and Miikkulainen, 2002), they have shown that, on different problems, such an exploration was much more efficient than a search driven by a distance towards a goal to be reached. This counterintuitive result has shown how strong the impact of the selection pressure is.

Several studies did propose to take into account a space that is specific to ER, i.e. the behavioral space, in the diversification phase. Trujillo et al. (2011) proposed a speciation mechanism based on behavior, while Gomez (2009) and Mouret and Doncieux (2009b,a) proposed to use behavioral distances for diversity preservation. Mouret and Doncieux (2012) made several comparisons with the following conclusions: (1) explicitly encouraging behavioral diversity leads to substantial improvements (2) multi-objective approaches lead to better results.

The impact of selection pressure on the evolution of cognitive abilities has been seldom studied. Capi and Doya (2005b) have shown that an evolutionary algorithm inspired from island models facilitates the evolution of memory, thus suggesting that the selection pressure has an impact on cognitive ability evolution. The goal of this paper is first to confirm the importance of selection pressure to the evolution of memory, and to propose fitness functions that promote this evolution.

Methods

The multi-objective approach has an interesting feature: adding a selection pressure can be done simply by adding it as a separate objective with no need to tune any new parameter for the relative importance of each objective to be optimized. This means that all objectives are considered equally important and multi-objective evolutionary algorithms aim at finding the best trade-off solutions relative to them (Deb, 2001). The two selection pressures studied here are then defined as separate objectives to be optimized with a multi-objective evolutionary algorithm. Such objectives do not describe the goal to be reached, but aim at enhancing the evolutionary search, they are then *helper objectives*. This approach is called multiobjectivization (Knowles et al., 2001; Mouret, 2011).

In the following, two helper objectives have been considered:

- a behavioral diversity, as defined in (Mouret and Doncieux, 2012);
- a scenario-based objective, as introduced in this work.

Behavioral diversity

The behavioral diversity assumes a distance function $d_b(x, y)$ between the behaviors x and y in a population of N individuals. The diversity associated with individual x is then computed in the following way:

$$div(x) = \frac{1}{N-1} \sum_{y \neq x} d_b(x, y)$$

The behavioral distance d_b will be described in the Experimental Setup Section.

Scenario-based objective

The generic framework for a scenario-based objective is described in Figure 1. An individual is simulated over a collection of predefined scenarios. Its behavior on the different scenarios is stored (here the behavior of internal neurons is considered). The fitness value of the objective is derived from the comparison of those behaviors.

Scenario-based objectives promote individuals with a behavior consistent over a predefined set of scenarios — without explicitly describing a target behavior. For instance in

order to promote robustness to noise, individuals could be simulated on scenarios with various levels of noise. Individuals that have close behaviors in those scenarios should be rewarded, while individuals whose behaviors are strongly affected by noise should be punished.

Behaviors of an individual are compared and the scenario based objective will reward either their similarity or difference. The design of this objective actually consists in defining the scenarios and choosing whether the corresponding behavior should be similar or different one with another. By rewarding the similarity between behaviors or, in contrast, their difference, the scenario based objective encourages the emergence of a coherent behavior.

The definition of scenarios and comparisons depends on the considered task, and will thus be described in the next section.

Experimental Setup

T-Maze navigation task

The task is an extension of the “roadsign problem” (Ziemke and Thieme, 2002; Rylatt and Czarnecki, 2000): an agent starts off at the bottom of a T-shaped maze, encounters an instruction stimulus (e.g. a light) while moving along a corridor and, when it reaches the junction, it has to turn left or right, depending on which stimulus has been encountered (Figure 2).

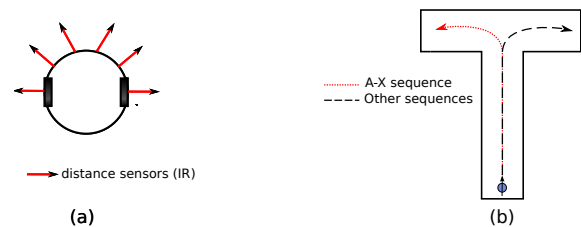


Figure 2: (a) Simulated mobile robot used for the T-maze task. The robot has four additional sensors, one for each letter. (b) Map employed for this task.

In the initial setup, controllers that simply follow the right or left wall after the signal can solve the task while not having any memory (Ziemke and Thieme, 2002). To make this task more cognitive, in our experiment the instruction stimulus is a combination of four stimuli (A, B, X, Y) following the same rule as in the AX-CPT working memory test (Braver et al., 1995; Pinville and Doncieux, 2010). This task consists of a context stimulus (A or B), followed by a second stimulus (X or Y) after some delay. The agent must turn to the left when the stimulus A is followed by the stimulus X, and to the right otherwise (for AY, BX, BY).

Here, the agent is a simulated two-wheeled robot receiving sensory inputs from 6 infrared distance sensors and four letter sensors, one sensor for each letter A, B, X, Y, which receives 1 if the letter is presented, 0 otherwise. The robot

controls its speed through two output units corresponding to its left and right motors. The agent is evaluated on each letter sequence (A followed by X, AY, BX, BY). The fitness increases by 1 if it turns to the correct side for the sequences AY, BX, BY and by 3 for the sequence AX, for a maximal value of 6. This fitness will be referred as “Goal oriented fitness”.

Both motors are disabled during the presentation of the letters. The whole task lasts 350 steps and takes place as follows with t the number of elapsed time steps:

- $0 < t < 50$: presentation of the first letter (A/B);
- $50 \leq t < 100$: delay, all the sensors are set to 0;
- $100 \leq t < 150$: presentation of the second letter (X/Y);
- $150 \leq t \leq 350$: the robot can move and must reach the correct side of the T-maze.

In order to avoid overfitting to a specific initial configuration of the robot, 12 different contexts have been defined for each possible letter sequence. A context is described by an initial starting position (4 different positions) and an initial starting angle (3 different angles).

The behavioral distance d_b between two individuals used to compute the behavioral diversity is the euclidian distance between the positions of the two robots at $t = 350$.

Neural network encoding

The agent is controlled by a neural network whose structure and parameters are evolved. DNN, a simple direct encoding inspired from NEAT (Stanley and Miikkulainen, 2002) has been used (Mouret and Doncieux, 2009b,a). It does not use crossover. Mutations can change parameters (connection weights and neuron biases) and add or remove neurons or connections. A IPDS-based (locally Projected Dynamic System) neuron model (Girard et al., 2008) is used to simulate the neurons with an output in $[-1, 1]$. It corresponds to a variant of the classic leaky integrator with similar dynamics but with the dynamic property of contraction (Girard et al., 2008). The same setup has already been used in (Pinville et al., 2011).

Scenario-based Objective

Each individual is simulated and evaluated on the 12 different contexts. In each of these contexts, one individual is simulated over the 4 different scenarios AX, BX, AY, and BY.

An individual has N internal neurons — N may vary from individuals to individuals and during evolution. For each scenario s , $b_s^i(t)$ is the output of the i -th internal neuron in scenario s at time-step t , after the presentation of letters ($t > 150$). The goal of the scenario based objective is to rewards individuals that obey the following rules:

$$\forall s \in S, b_{AX}^i(t) \neq b_s^i(t)$$

$$\forall s, s' \in S, b_s^i(t) = b_{s'}^i(t)$$

With $S = \{BX, AY, BY\}$. In other words, the behavior of an internal representation should be the same if the inputs are AY, BX, BY, and different if the input is AX. The behavior is computed after the presentation of letters, which means that the input letters are no longer active. The existence of a difference between the scenarios should reflect the emergence of a memory.

For each internal neuron i two partial fitness f_1^i and f_2^i are computed, they measure how well the internal neuron respects the two previous rules:

$$f_1^i = \frac{1}{|S|} \sum_{s \in S} \frac{1}{200} \sum_{t=150}^{350} \frac{|b_{AX}^i(t) - b_s^i(t)|}{2}$$

$$f_2^i = 1 - \left[\frac{1}{|S|^2 - |S|} \sum_{s, s' \in S, s \neq s'} \frac{1}{200} \sum_{t=150}^{350} \frac{|b_s^i(t) - b_{s'}^i(t)|}{2} \right]$$

Then, the fitness of each internal neuron is computed as follows:

$$f^i = f_1^i + f_2^i$$

As the goal of this experiment is to select individuals that have *at least* one internal neuron that represents the information, the final fitness is computed as the maximum of all internal fitnesses f^i :

$$f = \max_{0 \leq i < N} f^i$$

The fitnesses f compare the four letter sequences evaluated in the same context. The overall scenario-based fitness corresponds to the average of the 12 fitnesses thus defined (one for each context).

Setups summary

Throughout the article, we will refer to the different objectives as follows:

- **G**: Goal-oriented objective;
- **D**: Diversity objective;
- **S**: Scenario-based objective;

To test the influence of each objective, experiments are launched with various combination of objectives as shown in Table 1. The multi-objective evolutionary algorithm is NSGA-II (Deb, 2001) and each of these setups is run 30 times.

Results

Figure 3 depicts boxplots for the goal-oriented fitness results on each different setups. The red line represents the median value, the box extends from the lower to upper quartile values of the data. Flier points are those past the end of the whiskers.

Table 1: Summary of different setups used

	Setup	Description
1	G	Goal-oriented
2	G + D	Goal-oriented + Diversity
3	G + S	Goal-oriented + Scenario-based
4	G + D + S	Goal-oriented + Diversity + Scenario-based

Table 2 displays the corresponding p-values using Mann-Whitney statistical test. Figure 4 shows the median fitness values for the 4 setups.

Diversity Effect

Figure 3 shows that a simple fitness rewarding the completion of the task (G) has poor results. This is confirmed by Figure 4 in which one can see that a fitness plateau is quickly reached. The fitness plateau is at $f = 0.5$, which corresponds to controllers that always go to the same side of the maze. Adding a diversity objective (D) significantly increases performance and delays fitness plateaus. This result is compatible with our hypothesis that the evolution of a memory is a deceptive problem and shows that selective pressures have indeed a significant impact on the success rate.

Scenario-Based Objective Effect

The use of the Scenario-based Objective also increases the performance significantly, to the same extent as the diversity objective. There is no statistical difference between $G + D$ and $G + S$ setups.

Using both objectives further increases performance, and as no fitness plateau was reached during the 2000 generations (Figure 4). One can then expect the fitness to be even better with more generations.

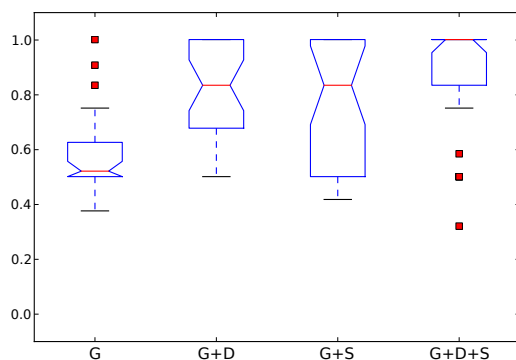


Figure 3: Boxplots for the goal-oriented fitness results on each different setups

The two next sections present a more in depth study of results: the resulting networks are tested for reliable memory

Table 2: P-values between each setup on goal-oriented fitness value

	G + D + S	G + S	G + D	G
G + D + S	x	0.04013	0.0639	<1e-05
G + S	0.04013	x	0.18504	0.00409
G + D	0.0639	0.18504	x	<1e-05
G	<1e-05	0.00409	<1e-05	x

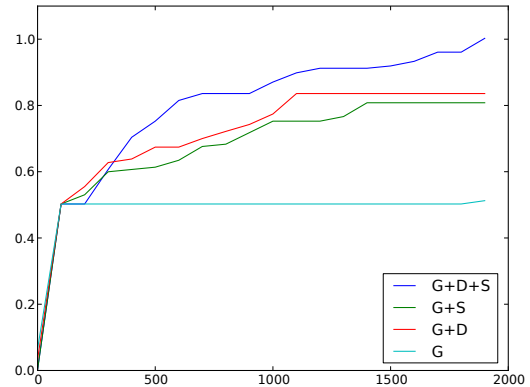


Figure 4: Evolution of fitness objective (median value of all 30 runs).

and generalization ability.

Memory computation A network is considered to exhibit a reliable memory if at least one internal neuron respects the two following points:

- After presentation of the letters, the neuron has a different output for AX scenarios and for AY, BX, BY scenarios.
- The memory is not affected by the duration of the presentation of the letters. While during evolution the duration of the presentation was 50 time-steps for each letter, the activity of the network is tested —after evolutionary process— with a duration of 400 time-steps. This is aimed to detect networks that rely on complex dynamics to have different activities after exactly 50 time-steps, but would not work with a different duration.

In Figure 5, the black histogram displays the percentage of runs (out of the 30 runs per setup) in which the best individual achieves reliable memory. While diversity objective slightly increases memory, the Scenario-based objective significantly affects memory. Interestingly using both helper objectives results in less memory than using the Scenario-objective alone.

Generalization ability Another important aspect studied here is the generalization ability. *During evolution*, the robot is tested in 12 different contexts for each letter sequence, and maximal fitness is achieved only if the individual manages to solve the problem in all the contexts. *After evolution*, the best controllers are tested in 180 previously unseen

contexts. The 180 new contexts include different map sizes, starting positions, and starting orientations of the robot. A controller is considered to generalize well if it can still perform the task in at least 60 of these new contexts. Figure 5 shows the proportion of runs with individuals which generalize. Figure 6 details the number of context in which these individuals generalize. There is a very significant increase of generalization when using the helper objective, and even more when using both objectives. Table 3 displays the corresponding p-values.

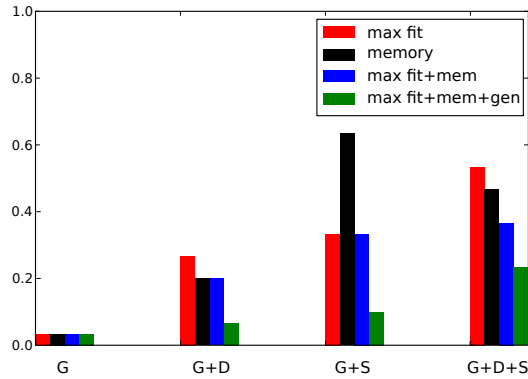


Figure 5: Proportion of runs matching different criteria: (1) achieving maximal fitness (2) having memory (3) having both (4) having both and generalizing to 60 of the 180 extra contexts.

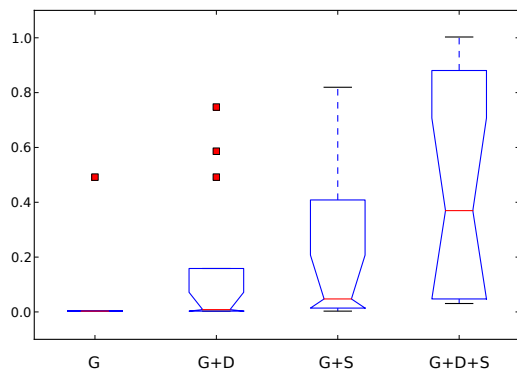


Figure 6: Generalization ability of the 15 best runs for the 4 different setups. The value corresponds to the normalized number of contexts in which the agent solves the task.

Analysis of the resulting networks

Two resulting networks, shown in figures 7 and 9, are analysed in this section. They both achieve maximal fitness, but

Table 3: P-values between each setup on the generalization ability

	G + D + S	G + S	G + D	G
G + D + S	x	0.03241	0.00364	1e-05
G + S	0.03241	x	0.08408	9e-05
G + D	0.00364	0.08408	x	0.0094
G	1e-05	9e-05	0.0094	x

only the second one exhibits reliable memory and generalization ability. Blue neurons have a different neural activity for AX sequence than the others during the memory test. Figure 8 and 10 show the corresponding internal behavior of the neurons during the test for networks in figure 7 and 9. The first presentation of letters lasts from 0 to 400, the delay from 400 to 800, the second letter from 800 to 1200. In order to distinguish AX and BX sequences, the network must remember A or B stimulus during the delay period.

In figure 8, we can see that the network depicted on figure 7 is not able to retain A or B stimulus when the delay interval is extended. At timestep $t = 800$, the internal behavior of the neurons are similar for the 4 sequences. At the end of the presentation of letters, the neural network cannot therefore distinguish AX and BX sequences. In figure 10, there are two different neurons, neurons 0 and 3, able to memorize stimulus B even if the delay interval is extended. In this case, at the end of the presentation of letters, the internal behavior of the neurons for AX sequence is different than for the other sequences.

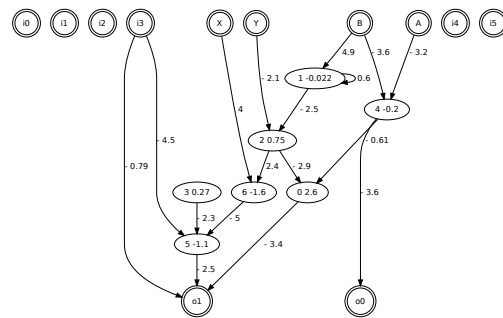


Figure 7: Resulting neural network with maximal fitness, but no memory nor generalization

Conclusion

These experiments confirm that the emergence of memory is a challenging problem. With the present encoding, structures with memory require several mutations to appear, will be much more likely to appear under specially-designed selective pressures. The helper objectives considered, both diversity and the newly defined scenario-based objective, significantly increase the convergence rate on this task.

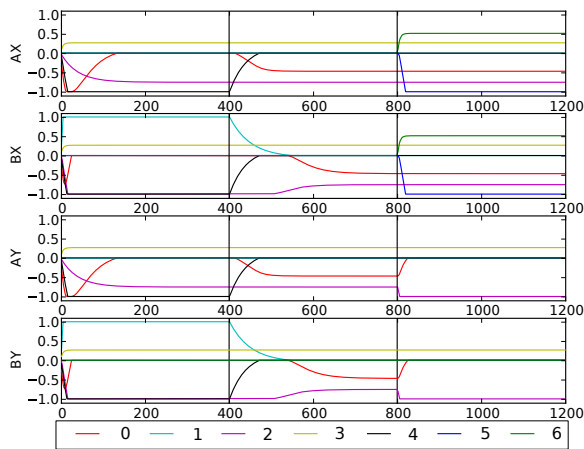


Figure 8: Internal behavior of the neurons corresponding to neural network displayed in Figure 7, for the 4 different sequences during the memory test.

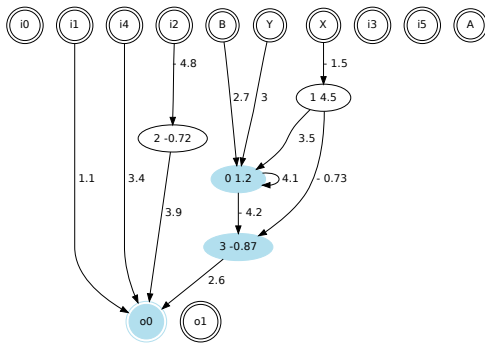


Figure 9: Resulting neural network with maximal fitness, memory and generalization

The scenario-based objective—and, to a lesser extent, the diversity objective—promote memory in the resulting networks. Moreover, the helper objectives are shown to have a large impact on generalization ability even though they aren't specifically designed to do so. We can hypothesize that there is a link between the presence of memory in agents and the generalization ability on this task.

The scenario-based method does not assume a specific structure and could potentially be used in any neuroevolution experiment involving elementary memory. The scenario-based objective is crucial here because it can select individuals with many different internal representations. Another methodological aspect highlighted in this paper is the use of a multi-objective evolutionary algorithm. Additional objectives are simply added, selecting individuals that might have a low fitness regarding to the main objective, but have an original behavior or efficient internal representa-

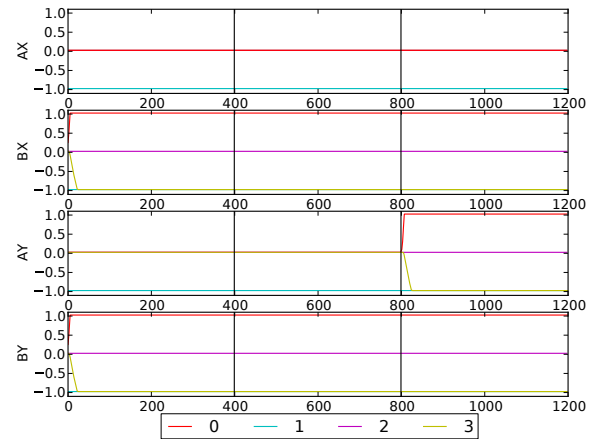


Figure 10: Behavior of internal neurons corresponding to neural network displayed in Figure 9, for the 4 different sequences during the memory test.

tion. We believe that those individuals can be good stepping stones to efficient cognitive solutions.

Future work The use of specific helper objectives and behavioral diversity objectives have a critical impact on the success rate of the presented experiments. However, Figure 5 shows room for improvement. Novelty Search (Lehman and Stanley, 2008, 2011) may also be defined as an helper objective (Mouret, 2011) and may thus be compared to the selection pressures proposed here. It should also be noted that the scenario-based method is not specific to the task nor the encoding. It could be applied to any neuroevolution encoding, such as NEAT (provided that it is adapted to multi-objective problems), or to fixed structures such as Elman or Echo State Networks.

Acknowledgments

This project was funded by the ANR EvoNeuro project, ANR-09-EMER-005-01.

References

- Allen, B. and Faloutsos, P. (2009). Complex networks of simple neurons for bipedal locomotion. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4457–4462. IEEE.
- Blynel, J. and Floreano, D. (2003). Exploring the T-maze: Evolving learning-like robot behaviors using CTRNNs. *Applications of Evolutionary Computing*, pages 593–604.
- Braver, T. S., Cohen, J. D., and Servan-Schreiber, D. (1995). A computational model of prefrontal cortex function. *Nips*, pages 141–148.
- Capi, G. and Doya, K. (2005a). Evolution of Neural Architecture Fitting Environmental Dynamics. *Adaptive Behavior*, 13(1):53–66.

- Capi, G. and Doya, K. (2005b). Evolution of recurrent neural controllers using an extended parallel genetic algorithm. *Robotics and Autonomous Systems*, 52(2-3):148–159.
- Deb, K. (2001). *Multi-objectives optimization using evolutionary algorithms*. Wiley.
- Dennett, D. C. (1997). *Kinds of Minds: Towards an Understanding of Consciousness*. Basic Books.
- Doncieux, S., Mouret, J.-B., Bredeche, N., and Padois, V. (2011). Evolutionary Robotics: Exploring New Horizons. In *New Horizons in Evolutionary Robotics: post-proceedings of the 2009 EvoDeRob workshop*, pages 3–25. Springer.
- Durand, N., Alliot, J.-M., and Medioni, F. (2000). Neural Nets trained by genetic algorithms for collision avoidance. *Applied Artificial Intelligence*, 13(3).
- Floreano, D., Dürr, P., and Mattiussi, C. (2008a). Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1(1):47–62.
- Floreano, D., Husbands, P., and Nolfi, S. (2008b). Evolutionary Robotics. In Siciliano, B. and Khatib, O., editors, *Handbook of Robotics*, pages 1423–1451, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Floreano, D. and Urzelai, J. (2000). Evolutionary robots with on-line self-organization and behavioral fitness. *Neural Networks*, 13(4):431–443.
- Floreano, D. and Urzelai, J. (2001). Evolution of plastic control networks. *Autonomous Robots*, 11(3):311–317.
- Girard, B., Tabareau, N., Pham, Q. C., Berthoz, A., and Slotine, J. J. (2008). Where neuroscience and dynamic system theory meet autonomous robotics: a contracting basal ganglia model for action selection. *Neural Networks*, 21(4):628–641.
- Gomez, F. J. (2009). Sustaining diversity using behavioral information distance. In *Proc. of GECCO'09*, pages 113–120. ACM.
- Heschl, A. (1990). $L = C$ a simple equation with astonishing consequences. *Journal of Theoretical Biology*, 145(1):13–40.
- Knowles, J. D., Watson, R. A., and Corne, D. W. (2001). Reducing Local Optima in Single-Objective Problems by Multi-objectivization. *First International Conference on Evolutionary Multi-Criterion Optimization*, 1993:268–282.
- Lehman, J. and Stanley, K. O. (2008). Exploiting Open-Endedness to Solve Problems Through the Search for Novelty. In *Artificial Life*, volume 11.
- Lehman, J. and Stanley, K. O. (2011). Abandoning objectives: evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223.
- Maturana, H. R. and Varela, F.-J. (1980). *Autopoiesis and cognition: the realization of the living*.
- Mouret, J.-B. (2011). Novelty-based multiobjectivization. In *New Horizons in Evolutionary Robotics: Extended Contributions from the 2009 EvoDeRob Workshop*, pages 139–154. Springer.
- Mouret, J.-B. and Doncieux, S. (2009a). Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In *Proc. of CEC 2009*, pages 1161–1168.
- Mouret, J.-B. and Doncieux, S. (2009b). Using Behavioral Exploration Objectives to Solve Deceptive Problems in Neuroevolution. In *Proc. of GECCO'09*, pages 627–634. ACM.
- Mouret, J.-B. and Doncieux, S. (2012). Encouraging Behavioral Diversity in Evolutionary Robotics: An Empirical Study. *Evolutionary computation*, 20(1):91–133.
- Nelson, A. L., Barlow, G. J., and Doitsidis, L. (2009). Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57(4):345–370.
- Pinville, T. and Doncieux, S. (2010). Automatic Synthesis of Working Memory Neural Networks with Neuroevolution Methods. In *Proc. of Neurocomp'10*.
- Pinville, T., Koos, S., Mouret, J.-B., and Doncieux, S. (2011). How to Promote Generalisation in Evolutionary Robotics : the ProGAb Approach Formalising the Generalisation Ability. In *Proc. of GECCO '11*, pages 259–266.
- Rylatt, R. M. and Czarnecki, C. A. (2000). Embedding Connectionist Autonomous Agents in Time: The 'Road Sign Problem'. *Neural Processing Letters*, 12(2):145–158.
- Stanley, K. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*.
- Tonelli, P. and Mouret, J.-B. (2011). On the Relationships between Synaptic Plasticity and Generative Systems. In *Proc. of GECCO '11*.
- Trujillo, L., Olague, G., Lutton, E., Fernández de Vega, F., Dozal, L., and Clemente, E. (2011). Speciation in Behavioral Space for Evolutionary Robotics. *Journal of Intelligent & Robotic Systems*, 64(3):323–351.
- Yamauchi, B. M. and Beer, R. D. (1994). Sequential Behavior and Learning in Evolved Dynamical Neural Networks. *Adaptive Behavior*, 2(3):219.
- Ziemke, T. (1999). Remembering how to behave: Recurrent neural networks for adaptive robot behavior. *Recurrent neural networks: Design and applications*, pages 341–376.
- Ziemke, T. and Thieme, M. (2002). Neuromodulation of reactive sensorimotor mappings as a short-term memory mechanism in delayed response tasks. *Adaptive Behavior*, 10(3/4):185–199.

Parameters

- MOEA: NSGA-II (pop. size: 200, number of generations: 2000)
- DNN (direct encoding):
 - prob. of changing weight/bias: 0.1
 - prob. of adding/deleting a conn.: 0.15/0.25
 - prob. of changing a conn.: 0.1
 - prob. of adding/deleting a neuron: 0.025/0.025
- Source code will be available online.

Evolved neural network controllers for physically simulated robots that hunt with an artificial visual cortex

Michael E. Palmer^{1,2} and Andrew Chou¹

¹Department of Biology, Stanford University, Stanford, CA 94305 U.S.A.

²mepalmer@stanford.edu

Abstract

Using a rule-based system for growing artificial neural networks, we have evolved controllers for physically simulated robotic "spiders". The controllers take their input from an "artificial retina" that senses other spiders and inanimate barrier objects in the environment, and must provide output to dynamically control the 18 degrees of freedom of the six legs of the robot every time step. We perform evolutionary runs with two species of spider that interact in simulation with each other and with inanimate barrier objects. One species (the "predator") is selectively rewarded for "eating" (by physically colliding with) the other species, and the other (the "prey") is selectively penalized for being caught, and rewarded for "eating" the barriers. The two species evolve complex running gaits, with control inputs coming from their retinas that produce hunting or avoidance behavior. We suggest that predator-prey frequency dependent selection can provide a relatively long-term genetic memory of previously searched regions of phenotype space, enforcing a form of novelty search that may reduce duplicated evolutionary search effort.

Introduction

One of the primary goals in the field of artificial developmental systems is to evolve systems that show an open-ended increase in complexity over evolutionary time. "Generative" artificial developmental systems (Boers and Kuiper, 1992; Jacob and Rehder, 1993; Gruau, 1994; Hornby and Pollack, 2001) are intended to provide the possibility of such open-ended increase, whereas, for example, a standard genetic algorithm with a fixed genome length, and fixed phenotypic meanings of all genetic loci, does not. We have previously (Palmer, 2011) made the observation that complexity is not necessarily selectively favored over simplicity: even though multicellular organisms exist, bacteria, in their relative simplicity, still make a fine living. Nonetheless, in natural history, it has apparently been the case that more complex organisms can sometimes do things that simpler ones cannot, and thereby outcompete them. If we can set up such situations, then we may be able to drive the evolution of complexity *in silico*. Gould (1994) has argued that the complexity of life *could* be due to drift; nonetheless, if we can create indirect selective pressure for it, complexity will increase much more rapidly than it would by drift alone.

Unfortunately, starting with primitive artificial organisms and immediately selecting for complex functions (such as complex cognition) typically results in uniformly low fitness

for all of the organisms. Long-term evolutionary progress requires a fitness landscape that offers selective "hints", on some timescale, pointing in the direction of greater fitness. These hints need not always be present, or even be fully consistent, but on evolutionary landscapes that sometimes reveal biases in their broader structure, an evolutionary algorithm can make long-term progress, rather than becoming trapped in local optima for long periods of time. This suggests that we can increase the probability of long-term evolutionary progress by creating a series of selective "stepping-stones" of increasing difficulty (for example, selecting for success at increasingly "difficult" cognitive tasks). This is also known as "incremental evolution" (Winkler and Manjunath, 1998) or "scaffolding" (Bongard, 2011). We might manually design a series of such stepping-stones on simple problems, but another long-term goal of artificial evolution is to solve problems that we do not know how to solve manually (or that are too expensive to solve manually). Our previous work on the "L-Brain" model (Palmer, 2011) defined a new generative method for "growing" neural networks via an artificial developmental process. Therefore we sought a means to automatically generate a series of increasingly difficult selective challenges, in an attempt to drive selection for complexity in such "grown" networks.

Predator-prey interactions have long been thought to promote adaptive evolution in artificial systems (Koza, 1991; Sims, 1994). A "Red Queen's race" between predator and prey may create an arms race of adaptation between them. A body of work by Nolfi et al. (Floreano and Nolfi, 1997; Nolfi and Floreano, 1998) and related work by Buason et al. (Buason and Ziemke, 2003; Buason et al., 2005) explored co-evolution of robot predators and prey, using a simulated version of a hardware robot. Our work in this paper differs in that we use a generative method for "growing" our neural network controllers, rather than evolving network weights only; however, some of the work by Buason (2003) involved the evolution of the robot bodies as well as their brains, which we do not discuss in this paper.

Whereas competitive co-evolution does sometimes produce interesting innovations, it can also, like single-species evolution, become stagnant. Alternatively, it can enter repeating "rock-paper-scissors"-like cycles, where predators and prey repetitively cycle through the same finite set of strategies to pursue, or evade, one another. Nolfi (2012) discusses these apparent obstacles to open-ended evolution in a review article, identifying some characteristics that promote

open-ended evolution; his answer can be generalized to say that “richness” of the evolutionary possibilities is important: he suggests that body-brain co-evolution contributes to this richness; as does “the ability of agents to adapt ontogenetically” or during their lifetimes, for example with memory; as does richness of the task and environment. It is our belief that generative developmental systems, as opposed to fixed genetic representations, may also add to the richness of evolutionary possibilities. In general, the larger the number of evolutionary avenues open to the combined co-evolutionary process of all the species, the less likely this process is to repeatedly tread the same evolutionary pathways over and over. (Clearly, the natural environment on Earth is vast and widely varying, especially if one includes the co-evolution of great numbers of competing species; so it does possess this qualitative “richness”.) In the work described here, we include four possible sources of evolutionary richness: 1) a generative developmental system for growing our neural network controllers, 2) a complex simulated robot body (which, herein, does not evolve, however), 3) two co-evolving species, a predator and prey, and finally, 4) we divide our population into a metapopulation of many local demes, with occasional migration between them, such that behaviors unique to a local “race” of one species might evolve in a subset of demes.

Methods

Robot Bodies and Actuation by Controllers

We use a fixed hexapod “spider” robot body (three of which are visible in Figure 1). Each of the six legs has three degrees of freedom (DOF): from the center of the head, looking outward along one of the upper leg segments, the “thigh” segment can move left-right and up-down. The attached “foreleg” segment can move up-down only. Neither joint can twist. Thus, in total, each robot has thirteen rigid body “parts”, and twelve joints with 18 total DOF, all of which are actuated. Each joint axis has fixed limits to its range of motion. The neural network controlling each robot body has one Output

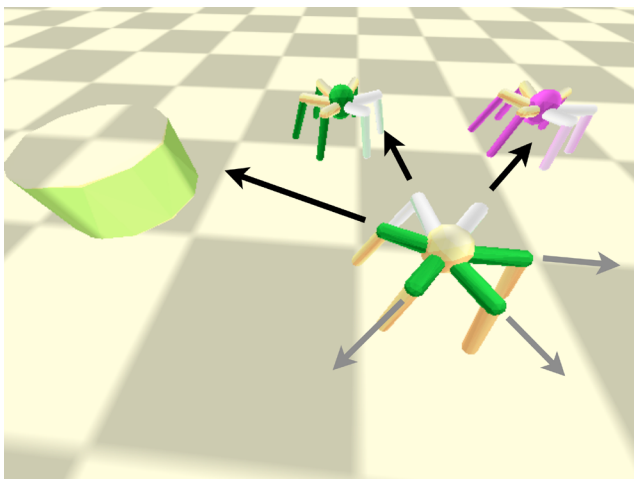


Figure 1: A focal spider sees three other objects in its environment along three of its six lines of sight (arrows).

neuron for each degree of freedom; when it assumes a value of +1, it is calling for its corresponding joint axis to be at its maximum range limit; a value of -1 calls for the minimum range limit. A simulated “spring” between the actual and requested positions generates a force on the joint axis.

We used similar robot bodies in (Palmer, 2011), but here we have eliminated the body orientation and velocity sensor Inputs, and replaced them with an “artificial visual cortex”.

Artificial Visual Cortex

An artificial “visual cortex” allows the spiders to “see” other objects of three types along six lines of sight. For example, in Figure 1, the green (which indicates the prey species) spider at lower right sees three objects situated around it: one spider of the same (green) species, one spider of the other (purple, indicating the predator) species, and one barrier object (cylinder). The lines of sight radiate from the spider at 60 degree angles; objects falling in a ~60 degree arc, centered on each line of sight, will register on the artificial visual cortex.

The 18 Input neurons of the visual cortex are arranged in three rows of six, as shown in Figure 2. A particular neuron activates when an object of its type is in a particular 60-degree arc (centered on one of the lines of sight): the rows encode the object type, and the columns encode the viewing direction. In Figure 2, three neurons in the visual cortex are activated (larger size), indicating the presence of one spider of the same species to the front left (“same2”), one spider of the other species to the front right (“other3”), and one barrier object to the left (“barrier1”), as they were situated in Figure 1. The neurons activate more strongly for closer objects.

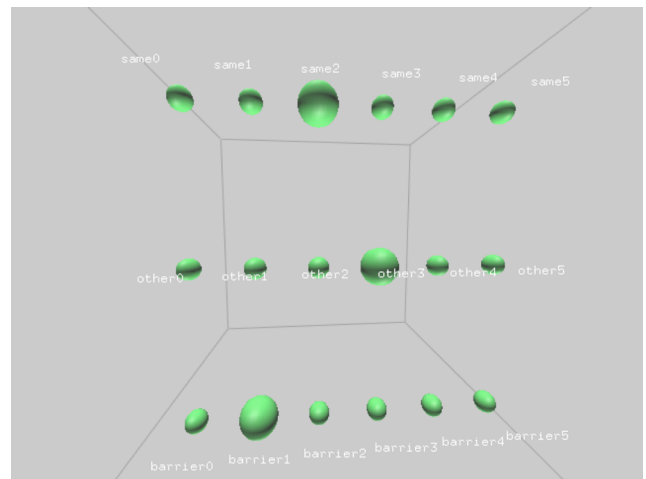


Figure 2: An “artificial visual cortex” registers the presence of three objects, indicated by the three large neurons.

Growth of Neural Networks

We use the L-Brain method (Palmer, 2011) for “growing” neural networks according to inherited sets of growth rules. In the L-Brain method, a neural network unfolds in three dimensions according to cell division rules comprising: 1) a predicate type, 2) a conditional expression that indicates when and where the rule may be applied, and 3) two successor

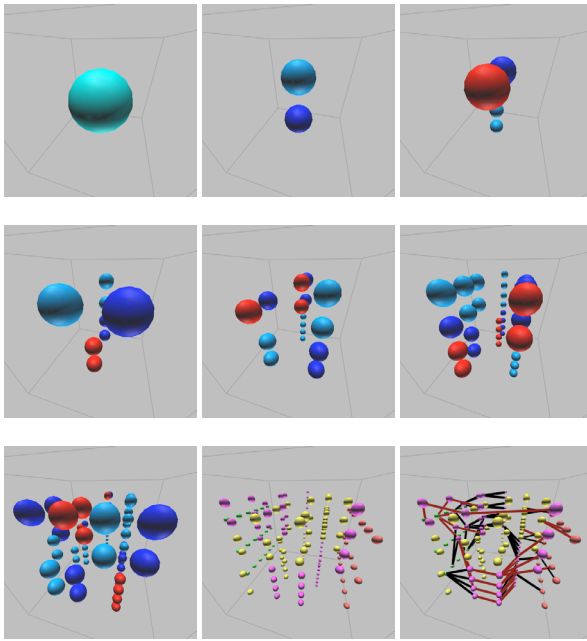


Figure 3: From a single initial protoneuron, seven division steps produce many protoneurons. Cell differentiation into types (colors), and the direction of division (X, Y or Z) is dictated by an inherited set of growth rules. Certain protoneurons convert into neurons, and red Outputs and green Inputs appear (bottom center panel). Synaptic connections form (bottom right panel).

types. Beginning from a single protoneuron of a certain type (indicated by the light blue color of the sphere in the top left panel of Figure 3), the rule set is repeatedly searched for applicable rules. If the predicate of a rule matches the type of a protoneuron, and the conditional expression evaluates to *true*, then the protoneuron divides into two protoneurons, each with one of the successor types (indicated by various sphere colors). The conditional expressions are intended to control neural development in a space-, time-, and context-dependent way, analogous to natural gene regulation. The expressions consist of a sequence of tokens of several types defining a Reverse Polish Notation (RPN) arithmetic expression, which operates on a set of four stacks (one stack of floating point values, one boolean stack, and two integer stacks; see (Palmer, 2011) for full details). As each token is evaluated in sequence, values may be popped from the stacks, specific operations performed on them (for example, two values might be summed, or tested for equality), and the result pushed back to a particular one of the stacks. Some tokens may push values onto the stacks that depend on time (the division number) or on the position of the neuron in space. After all the tokens in an expression have been evaluated, the stacks will in general hold a number of values. One of these values (the top value on the boolean stack) is used to determine whether the expression evaluates *true*, which permits a cell division to occur. Other values (from the floating point and boolean stacks) are used to specify parameters required by the neurons, for example, the weights applied to a neuron's inputs; see (Palmer, 2011) for

details. A maximum of seven divisions are applied; the direction in space of the division also depends on values from the boolean stack. The size of each protoneuron in the panels of Figure 3 indicates the step at which it stopped dividing.

At the bottom center panel of Figure 3, the final division occurs, and one additional application of the rules converts some of the final protoneurons into neurons of several classes, including Sigmoid (purple), Delay (cyan), and Oscillating (yellow). Sigmoid neurons sum their inputs plus a “bias” value, and apply a sigmoid normalization function to keep the output in the range $[-1, 1]$. Delay neurons take their input and buffer it for a certain number of time steps in a FIFO queue, then output it. Oscillating neurons oscillate sinusoidally between -1 and 1 over a fixed period; they have no inputs; see (Palmer, 2011). Also in the bottom center panel, a fixed set of 18 Input neurons (green, the “visual cortex” neurons from Figure 2) and 18 Output neurons (red, each of which will control one of the 18 DOF of the robot) are introduced.

In the bottom right panel of Figure 3, synaptic connections are formed. These also grow according to the inherited rule set: briefly, the final neurons have a set of “preferred” types to which they would like form connections to, and from; these preferred types are called “want-ins” and “want-outs”, and are supplied, respectively, by the two integer stacks. The final connections that are made satisfy a combination of these preferences with a locality requirement. The L-Brain method itself is not the focus of this paper, but see (Palmer, 2011) for much more detail. A video of the unfolding developmental process is available here: <http://www.youtube.com/alifespider>

Evolutionary Parameters

Predator and prey interactions. In (Palmer, 2011), we successfully used a single species to evolve a neural controller that would direct the 18 DOF of the robot to produce a “galloping” gait, and then track a compass heading to gallop to the North. In this paper, our goal is to study the interaction of two co-evolving species, one predator and one prey, selected for hunting and evasion behavior. The two species have identical body configuration and physical strength (maximum motor torque in each DOF), and their brains have identical growth constraints (same number of divisions, neuron types, etc.), but the two species are scored differently. A predator individual receives credit for “eating” a prey individual, by physically colliding with it; the prey is penalized for being eaten, and rewarded for eating inanimate barrier objects. (More details on scoring given below.)

Fitness evaluation in physically simulated local “demes”. We place $N=25$ individuals of each species into D separate demes (local populations), where D ranged from 16 to 320; thus the total metapopulation size is ND individuals, ranging from 400 to 8,000, of each species. Each individual has a distinct genotype, i.e., a distinct set of inherited rules. Both species are asexual. All the $2N$ robot bodies in a single deme are simulated together, along with N barrier objects; thus they may physically interact. Fitness is relative among all individuals of each species, within one deme. A single evaluation lasts for 2,000 time steps of 1/30 second each, for a total of just over 1 minute of simulated time. During this time,

robots accumulate a score at each time step, according to the details of the physical simulation, including the velocities of the robots, and whether collision events between objects occur. When a prey is “eaten”, it receives a score penalty, but does not disappear from the simulation; rather it is “regenerated” (retaining its accumulated score) in a new random location and the simulation proceeds. Similarly, when a barrier is eaten, it also moves to a new location. Individuals migrate to a new random deme at a rate of 0.01 per generation; which connects all demes into a large metapopulation. Thus one evolutionary generation consists of: 1) fitness evaluation via 2,000 time steps of physical simulation; 2) reproduction according to relative fitnesses; 3) possible mutation of the “rules” making up each genotype (at a rate of 0.05 per rule per generation); and 4) migration among demes. Evolutionary runs of 3,000 to 10,000 generations were typically performed.

Results

As cited above, it has long been suggested that competitive co-evolution can drive adaptive progress. Therefore, we had initially looked to co-evolution as a “magic bullet” that would provide an indefinite series of challenges, as the two species engaged in an arms race of adaptive improvement. However, we found that, in practice, a proper balance of selective forces was quite difficult to get right. When it seems that an interesting predator-prey interaction may be observed almost anywhere one looks in nature, it is actually the case that the present two-species interactions are themselves the result of a selective process. That is, we only observe those two-species interactions where both species have not died out, either due to extinction of the prey (when the predator is too efficient a hunter), or extinction of the predator (when the prey is too effective at escaping), or both. In our simulations, we do not allow either species to go extinct; we use a fixed population size and relative fitnesses. However, instead of extinction, a common result in our initial experiments was evolutionary stagnation; for example, that the prey would not evolve a forward running gait, because doing so causes them to risk blindly running into predators. Thus the initial part of our experiments was characterized by manual tuning of the scoring function, in a sequence of attempts to get the two species to evolve a galloping gait, and to interact, as follows.

Initial selection for forward motion. Our first experiments did not include the barrier objects, only the predators and prey, and we conducted them with $D=16$ demes of $N=25$, or $ND=400$ individuals of each species. Most initial randomly generated genotypes (L-Brain rule sets) produce no motion in the bodies they control; most do not even make any connections to their output neurons. However, a few genotypes may produce a wiggling motion in the robot, which may be improved gradually by evolution into a galloping gait, and by the ability to steer according to the inputs from the visual cortex, ultimately producing the ability to hunt or evade the other species. Although our focus was hunting and evasion behavior, we expected that some initial selection for basic forward motion would greatly speed up evolution of hunting, because stationary robots all fail to hunt, or evade, and thus

cannot be differentially rewarded for this behavior. Therefore, we initially included a positive bonus to reward forward motion in both species. The result was that both species would typically evolve a forward gait in less than 300 generations.

Introduction of barrier objects. We next introduced an additive reward for the predator, and an additive penalty for the prey, for captures of a prey by a predator. Unfortunately, if the penalty to the prey was strong, this causes the prey to evolve a dramatically reduced speed, or to stop; forward motion causes it to risk running into a predator, and slowing or stopping reduces this risk. To encourage the prey to continue forward motion, we introduced the barrier objects, and a reward to the prey for capturing them; this reward was large: four times the penalty of being caught by a predator. Even when a prey is running blindly (without use of input from the visual cortex), it receives a reward from time to time by blindly colliding with barriers, and over a 2,000 time step evaluation, this is less on average than the total penalty from colliding with predators.

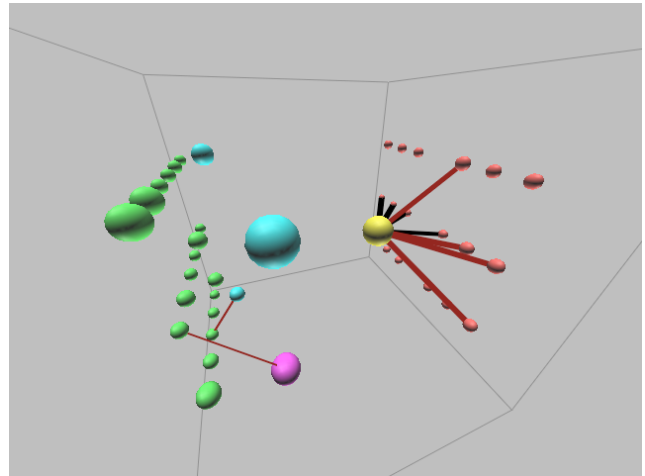


Figure 4: A “blind” galloper. The yellow Oscillator neuron at right produces a galloping gait by the pattern of pulsations induced in the red Output neurons. However, none of the green Input neurons has any pathway to any of the red outputs; this individual is blind.

Direct selection on network properties. At this point in our experiments, both predator and prey would easily evolve to run forward blindly, but their networks usually did not possess any connective pathways from the visual cortex Input neurons to any of the Output neurons that directly activate the legs. Such pathways are necessary for steering by visual cues. An example of such a “blind” brain is shown in Figure 4. The color and thickness of the lines connecting the neurons indicates the sign (black indicating positive sign, and red negative) and magnitude of their weight. In this network, when the yellow Oscillating neuron pulsates, the several connected red Output neurons also pulsate, either in phase if they are connected by a black line, or in opposite phase if connected by a red line, producing a pattern of movement in the legs that generates a running gait. The speed of the gait produced by this network is quite fast, but it does not steer

according to visual cues, because the Inputs do not pass a signal to the Outputs by any pathway.

We decided to select directly on properties of the neural networks. However, a bonus proportional to the total number of neurons was unproductive. A bonus proportional to the longest connected pathway was also unproductive. We found that a bonus that counted the total number of Input neurons that possessed some connective pathway to at least one Output did promote the evolution of running with visual steering. If *totalInputsConnected* is the number of Inputs that have some pathway downstream to some Output, then we compute the factor $F1 = \text{pow}(1.10, \min(5, \text{totalInputsConnected}))$, and multiply the total score by this factor. The factor F1 provides a 10% multiplicative reward for up to 5 Inputs that are connected by some path to an Output. This reward biases variation to the neighborhoods of networks that we desire, i.e., those that have Inputs somehow connected to Outputs. We subsequently also added an additional 2% multiplicative bonus $F2 = \text{pow}(1.02, \min(5, \text{longestPath}))$, where *longestPath* is the longest path present in the network. This bonus rewards networks with longer pathways (up to a length of 5) in order to speed the stage of evolution where complete pathways between the Inputs and Outputs are being found.

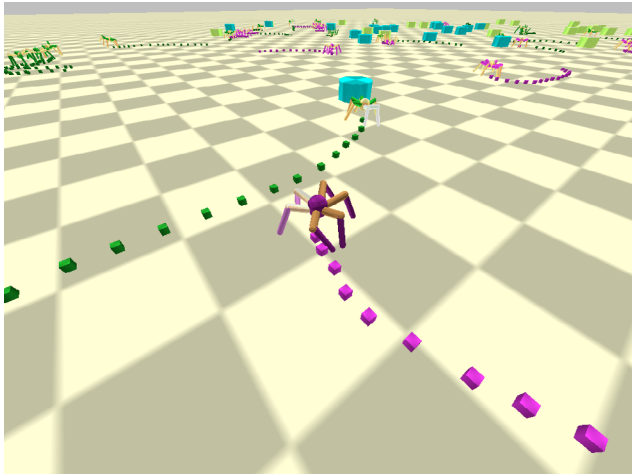


Figure 5: A predator (purple) tracking a prey (green), which is, in turn, tracking a barrier object.

Density of objects. At the beginning of an evaluation, all spider bodies and barrier objects are initially created in random positions within a circle of a certain radius. When objects are “eaten”, they are “regenerated” in a new position, with the new positions similarly distributed within the same circle. If a spider of either species ever runs outside the perimeter of the circle, it is also moved to a new interior location; this prevents the spiders from scattering so far that they cease to interact. We found that the density of spiders and objects imposed by this limiting circle was important: if they are too dense, then captures happen too easily by accident; if they are very sparse, then too few captures happen within an evaluation run of 2,000 time steps, so selection is noisy (i.e., too dependent upon the luck of being initially near a target). We arrived at a suitable radius for the containing circle for the N=25 spiders of each species and the 25 barriers,

i.e., about 50 simulation length units, where the span of the spider’s legs in a relaxed stance is 1.5 units; the resulting density can be seen in Figure 5.

Initial hunting success. With the above scoring function and object density, we began to have success evolving hunting behavior in both species, using N=25 and D=16. In Figure 5, an example of successful hunting behavior in both species is shown: a predator tracks a prey, which is itself tracking a barrier object. The spiders leave colored “breadcrumbs” behind them to make their recent track visible (however, the breadcrumbs are not visible to the spiders). A video of successful hunting behavior is available at: <http://www.youtube.com/alifespider>

One alternative outcome to the evolution of hunting in both species is that the prey may become faster runners (i.e., by developing a more efficient gait) than the predators, such that even a predator that is successfully tracking a prey cannot catch up; this in turn reduces the selective advantage to the predators of good tracking, and they cease to improve it, or may even lose it. (Thus, interestingly, predators that are good at tracking get more “practice”, and become better at it.)

“Orbit the barrier” baiting behavior. One alternative adaptive strategy taken by the predators, if a deme enters the slower predator / faster prey condition, is what we call predators’ “orbit the barrier” behavior. In some runs, the predators would circle around a barrier object, apparently waiting for prey to track toward the barrier. When the prey finally approaches and “eats” the barrier, it is not difficult for the predator to move into the center of the barrier (which has just disappeared, having been “eaten”), and capture the prey. In Figure 6, a predator circles a barrier as a prey approaches. A video of the “orbit the barrier” behavior is available at: <http://www.youtube.com/alifespider>

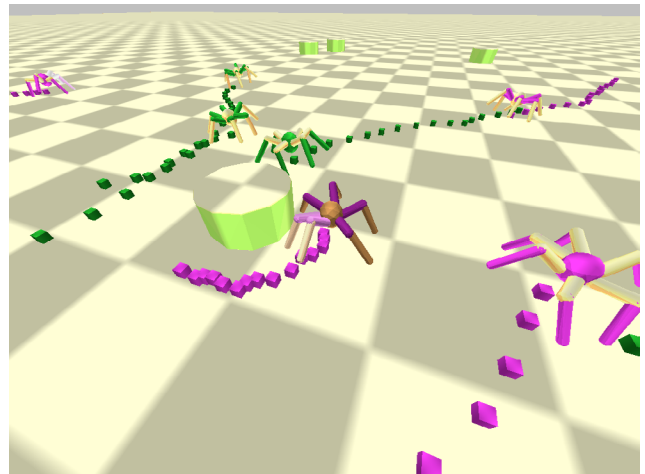


Figure 6: A predator (center, purple) engages in “orbit the barrier” behavior, waiting for prey (green), to approach.

Larger metapopulations. With D=16, not every run would produce successful hunting in both species. Using a single node of our 20-node computing cluster (each node contains 2 E5520 4-core CPUs), we are able to conduct a D=16 run on a

single cluster node at a rate of about 200 generations per hour, for $N=25$ (25 individuals of each species, and 25 barriers). In order to run larger metapopulations, we linked the 20 cluster nodes together by passing migrant individuals among them. That allowed us to run one large metapopulation of $D=20 \times 16=320$ demes (or $ND=8,000$ of each species) on the entire cluster, at the same rate of 200 generations per hour.

Runs with larger metapopulations produced additional refinements to behavior. The “circle the barrier” behavior we previously described first arises by predators blindly bumping into a barrier, and having a gait that does not allow them to disengage from it. In larger runs, we commonly see this being refined by predators that can visually track barriers, close on them, and then circle them. This appears to occur when the prey are already accomplished trackers: a “baiting” predator is relying on the prey’s tracking ability.

In these larger runs, we observed prey that shy away from predators: if one of these prey individuals is tracking towards a barrier, and the experimenter manually places a predator in its path, it will detect the predator (the appropriate “other” Inputs are connected, and activate) and divert its course, in order not to collide with the predator. Interestingly, we have also observed prey that will shy away from other prey (and we can see that the appropriate “same” Inputs are activated during this behavior). The adaptive value of this may be that two spiders that collide usually end up with their legs tangled together, which they often cannot disentangle, preventing them from running; two entangled prey are easy targets.

Interestingly, we have also, rarely, observed predators that track *toward* other predators, so that they collide with them; we are not sure whether this is adaptive or not. We have only observed it in the case of slow predators / fast prey, where the predators are also actively tracking the barriers; so it is possible that they benefit by tracking other predators when those predators are likely to already be circling barriers. If this behavior is in fact selected, it would be a third-order interaction, i.e., the prey are attracted to barriers; thus predators are attracted to the barriers; thus predators are attracted to other predators. This might be selective relative to being blind, but not relative to tracking the barriers directly.

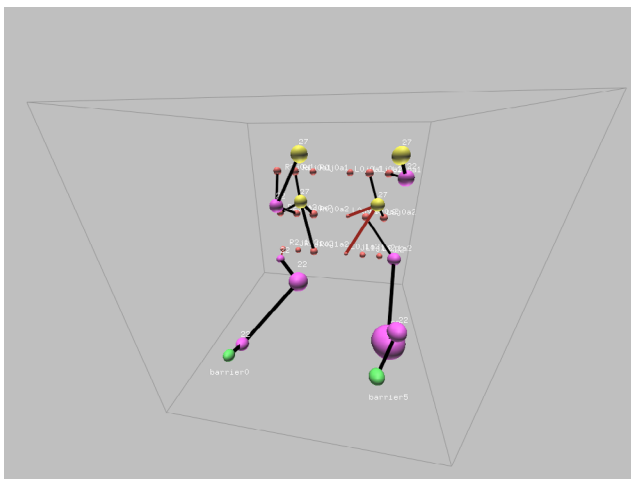


Figure 7: A (prey) brain that exhibits successful tracking behavior by zig-zagging toward a barrier object.

Typical brain structure for hunting behavior. A common structure for a brain (here, a prey) that exhibits successful tracking behavior is shown in Figure 7. Only neurons that are “upstream” of some Output are shown in the figure, because only those can affect the gait. Two green Input neurons are so connected, “barrier0” and “barrier5”. The typical behavior produced by such brains is to run in circles until a target object (a barrier in this case) appears (e.g., after being eaten and moved to a new location) near the spider. When one of the Inputs detects a target, the spider will turn left or right until it is moving toward the target. It zigzags back and forth as it closes on the target, with the target alternately activating the two Input neurons as it passes into their line of sight. Each activation causes a “zig” or a “zag” that diverts the path of the spider back toward the target, until it eventually closes on the target and “eats” it.

A common way this evolved tracking algorithm may fail is when two or more target objects are nearby on either side of the spider; this can cause tracking anomalies, such that capture fails. In addition, with moving targets (i.e., a prey being tracked by a predator), the target may move across one of the lines of site, and outside the “tracking cone”, also causing failure to capture. Commonly, a spider is under time pressure to quickly capture a target that it has sighted, lest a competing spider get to it first. Videos of spiders competing in this way are available at: <http://www.youtube.com/alifespider>

Complex evolutionary dynamics. The evolutionary dynamics created by this rich environment can be complex. In Figure 8, the prey (green points) actually slow down between generations 700 and 800 (top panel, Distance Covered) while they simultaneously increase Captures of barriers (middle panel). This is associated with an increase in the number of Inputs Connected to Outputs (bottom panel) by some pathway, indicating that they have made trade-off of speed for better tracking ability. This is associated with a mean drop in Captures (middle panel) by the predators (red points), as well as a fitness decrease (not shown), indicating that they were relying on the prey blindly running into them for some captures. Their apparent response in the short term is to speed up, and reduce their mean number of Inputs Connected.

It is not until generations 1100-1300 that the predators are able to increase their mean Captures again, not by increased speed, but apparently by better tracking, associated with a gradual increase in Inputs Connected to Outputs.

Conclusions

We have demonstrated a system that produces complex predator-prey dynamics, in a realistically modeled physical environment, with a generative developmental process producing the neural network controllers.

It turns out in practice that an “arms race” between predator and prey in artificial evolution is nontrivial to produce. For example, we initially encountered the situation where the prey would stop running, in order not to blindly run into predators, because it had not yet evolved an effective visual cortex. Thus we resorted to a number of *ad hoc* scoring changes, including direct selection for forward motion, and selection on network properties. In both cases, we are intelligently searching for evolutionary “stepping stones” to produce a particular result,

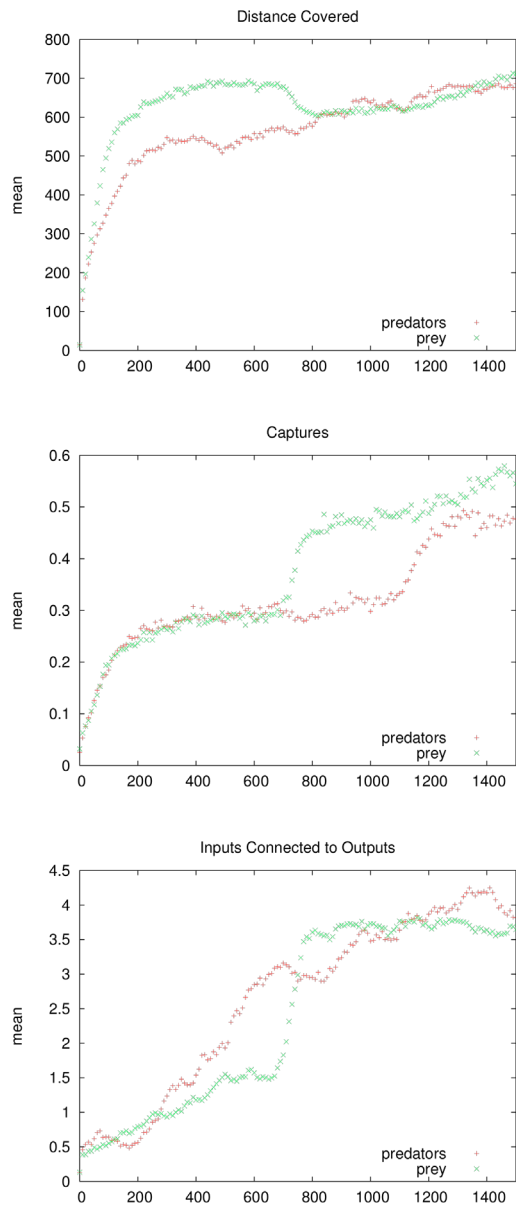


Figure 8: Trade-offs create complex dynamics.

tracking behavior in this case. Presumably, the evolutionary process, even without such hints, (e.g., with selection only including a reward for hunting and a penalty for being hunted) would *eventually* find a solution (even by drift alone), but this could take far longer. Thus, ironically, in order to produce a rich predator-prey interaction, in the hope that this would produce open-ended evolution, we found it necessary to start the process with “scaffolding” hints.

One interesting effort that aims to make evolution more efficient is called “novelty search” (Lehman and Stanley, 2011), which keeps track of the regions of phenotypic space that have been previously searched, and does not produce similar organisms again. However, when phenotypic space becomes very multidimensional, it may not be straightforward to characterize and record the previously searched volume of

phenotype space; nor is it clear how to reduce the dimensionality of this representation (to compress, and to search, the record) in general. For example, if the goal were to produce increasingly efficient running gaits in a single species, it might initially be sufficient to penalize individuals that exhibit the non-novel behavior of standing still (which many randomly generated rule sets produce). However, later on, when many individuals are running, this may be insufficient if the population settled into a local optimum. We might have to identify and measure many subtle aspects of running gaits to identify which behaviors we should call “similar”, in order to reward novelty. We suggest that simple, static novelty metrics will fail to “scale up”, in the sense of continuing to produce improvement. In general, the problem of defining an increasingly complex novelty metric will be similarly difficult to defining a sequence of evolutionary stepping stones.

Although evolutionary computing does not commonly employ a fitness that is negatively frequency-dependent (a fitness that decreases as the frequency of that phenotype increases), many natural processes produce such selection; “apostatic selection” is selection that favors individuals deviating from the norm (Ayala and Campbell, 1974).

We note that predator-prey interactions do force a temporally-local “novelty search” due to the apostatic selection of predator-prey interactions: when the predator adopts strategy A, and the prey adopts strategy B to counter it, then *at least for a short time*, the predator is forced to find a novel, non-A solution. Importantly, no manual dimensionality-reduction of the phenotype space is required: the discouraged strategy (A) is encoded, in a sense, in the genome of the prey. When the predator changes to another strategy, and the prey follows, then this “memory” of the previously covered region of phenotype space is lost – or is it? It is possible for second-order selective effects related to evolvability (Wagner and Altenberg, 1996), i.e., the tendency to *produce* adaptive variation, to shape the genome: even through the prey is no longer currently expressing the B strategy, its genome may now be more easily able to re-evolve the B strategy. This produces a “memory” on a longer timescale: if the predator adopts A again, it may be more quickly countered with B. Predators that evolve a novel, non-A strategy, can thus be rewarded on this longer timescale. Only when A has been avoided – and novelty has been enforced – for a very long time may this “memory” eventually fade. A similar dynamic may occur when species compete for limited resources: when one resource is overexploited, novel use of available resources is favored. Good “resource switchers” may be favored in the long term. Similarly, not only are specific predation behaviors selected in the short term, but the general ability to evolve among a range of predation behaviors, in response to locally prevalent prey counter-strategies, may be favored in the long term.

This article has been a largely qualitative demonstration of the ability of our system, given some encouragement by scaffolding, to produce complex predator-prey interactions. The system does successfully produce third-order interactions (predators tracking other predators, which track barriers, which are tracked by prey; this increases the chance that the first predator type collides with prey). This clearly adds to the “richness” of behavioral interactions many levels removed

from the mutation of rules in the genotype. Our intention now is to use this platform to study how this richness and diversity of phenotype can be made indefinitely self-sustaining.

Acknowledgements

Many thanks to Marc Feldman and the members of the Feldman lab at Stanford University. This work was supported in part by NIH grant GM28016.

Appendix

There is not space here to provide sufficient details for repeatability of these experiments, since our system is large and complex; therefore we will be releasing our source code online at the following URL: <http://mepalmer.net/alifespider>

We have made a number of changes and improvements to the original L-Brain system described in (Palmer, 2011). We list the major ones briefly here: 1) We adjust the raw fitnesses by adding a quantity to all raw fitnesses such that the 75% fitness percentile individual has 4x the adjusted fitness value of the 25% fitness percentile individual. This has the effect of intensifying selection when the variance in the raw fitness is low. 2) We increased the possible number of types of protoneurons from 18 to 21, and the numbers of possible types of Oscillating, Sigmoid, and Delay neurons from 6 to 7 of each. 3) In the original paper, each L-Brain rule possessed a single predicate; here, an L-Brain rule possesses two possible predicates; if either alternate predicate matches the neuron type, then the rule applies; this causes more divisions to succeed, producing denser networks, in the initial random rule sets. 4) Each neuron originally had a single preferred “want-in” and “want-out”, the types it “prefers” to receive connections from, and send connections to, respectively; here, each neuron receives a ranked list of 4 want-ins and want-outs; a combination of preference order and locality constraints is considered to determine the connections that are finally made. 5) In the original paper, we began with three initial protoneurons; here we begin with a single one. 6) The physical simulation has been sped up by increasing the time step size from 1/60 to 1/30 sec, with negligible error. 7) We introduced three “stages” of development, and each rule is marked as only applying during one of the stages: a) initial protoneuron development, b) conversion of protoneurons into neurons, and c) computation want-ins, want-outs, and other parameters for Input and Output neurons; previously, the Inputs and Outputs did not have want-in and -out preferences. 8) Additional scoring corrections: spiders get no credit for hunting if they are upside down, and we add an additional term computed by dividing the number of captures by the velocity, in order to reward efficient hunting.

References

- Ayala, F. J., and C. A. Campbell. 1974. Frequency-Dependent Selection. *Annual Review of Ecology and Systematics* 5:115-138.
- Boers, E. J. W., and H. Kuiper. 1992. Biological metaphors and the design of modular artificial neural networks. *Departments of Computer Science and Experimental and Theoretical Psychology. Leiden University, the Netherlands.*
- Bongard, J. C. 2011. Morphological and Environmental Scaffolding Synergize when Evolving Robot Controllers. Pp. 179-186. *Genetic and Evolutionary Computation Conference (GECCO)*. ACM, Dublin, Ireland.
- Buason, G., N. Bergfeldt, and T. Ziemke. 2005. Brains, Bodies, and Beyond: Competitive Co-Evolution of Robot Controllers, Morphologies and Environments. *Genetic Programming and Evolvable Machines* 6:25-51.
- Buason, G., and T. Ziemke. 2003. Competitive Co-evolution of Predator and Prey Sensory-Motor Systems. Pp. 605-615 in S. Cagnoni, ed. *EvoWorkshops 2003*. Springer-Verlag.
- Floreano, D., and S. Nolfi. 1997. Adaptive behavior in competing co-evolving species. Pp. 378-387. *Proceedings of the Fourth European Conference on Artificial Life*. MIT Press.
- Gould, S. J. 1994. The Evolution of Life on Earth. *Scientific American* October 1994:85-91.
- Gruau, F. 1994. Neural Network Synthesis Using Cellular Encoding and the Genetic Algorithm. *Computer Science. l'Ecole Normal Supérieure de Lyon*.
- Hornby, G. S., and J. B. Pollack. 2001. Body-Brain Co-evolution Using L-systems as a Generative Encoding. Pp. 868-875. *Genetic and Evolutionary Computation Conference (GECCO 2001)*, San Francisco, CA.
- Jacob, C., and J. Rehder. 1993. Evolution of Neural Net Architectures by a Hierarchical Grammar-based Genetic System. Pp. 72-79. *Artificial Neural Nets and Genetic Algorithms (ANNGA) '93*.
- Koza, J. R. 1991. Evolution and Co-Evolution of Computer Programs to Control Independently-Acting Agents. *Proceedings of the First International Conference on Simulation of Adaptive Behavior*. MIT Press.
- Lehman, J., and K. O. Stanley. 2011. Abandoning Objectives: Evolution through the Search for Novelty Alone. *Evolutionary Computation* 19:189-223.
- Nolfi, S. 2012. Co-evolving predator and prey robots. *Adaptive Behavior* 20:10-15.
- Nolfi, S., and D. Floreano. 1998. Co-evolving predator and prey robots: Do “arm races” arise in artificial evolution? *Artificial Life* 4:311-335.
- Palmer, M. E. 2011. Evolved Neurogenesis and Synaptogenesis for Robotic Control: The L-brain Model. *Genetic and Evolutionary Computation Conference (GECCO 2011)*, Dublin, Ireland.
- Sims, K. 1994. Evolving 3D Morphology and Behavior by Competition. Pp. 28-39 in R. Brooks, and P. Maes, eds. *Artificial Life IV Proceedings*. MIT Press.
- Wagner, G. P., and L. Altenberg. 1996. Complex adaptations and the evolution of evolvability. *Evolution* 50:967-976.
- Winkeler, J. F., and B. S. Manjunath. 1998. Incremental Evolution in Genetic Programming. Pp. 403-411. *Genetic Programming 1998: Proceedings of the Third Annual Conference*. Morgan Kaufmann.

Evolution of Virtual Creature Foraging in a Physical Environment

Marcin L. Pilat^{1*}, Takashi Ito, Reiji Suzuki and Takaya Arita

Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
¹pilat@alife.cs.is.nagoya-u.ac.jp

Abstract

We present the results of evolving articulated virtual creature foraging in a 3D physically simulated environment filled with stationary food objects. Simple block creatures with sigmoidal neural networks are evolved through a genetic algorithm using a fitness function based on the consumption amount. The results show the evolution of successful foraging behaviors performing well in environments with various food distributions. We analyze the foraging based on its efficiency, creature morphologies, movement strategies, and the food density and entropy in the simulation environment.

Introduction

Movement plays a crucial role in the fate of most biological organisms and is the theme of active and diverse research in biology (Holyoak et al., 2008). Morphologies constrain the movement of organisms allowing them to find food, escape predation, and reproduce. Thus, they are of crucial importance for organism survival. Studying morphology and development, especially in the context of ecology, will contribute to answering difficult biological challenges and promises direct applications to society (Wake, 2001).

We are interested in the evolutionary study of exploratory movement through physical simulation in the context of movement ecology. The movement ecology paradigm (Nathan et al., 2008) is a conceptual framework for the study of organismal movement promising to enhance our understanding of the causes, mechanisms, and consequences of movement in the biological world. Physical simulation provides an ideal framework for studying the evolution of functional morphologies and the movement they enable.

We hypothesize that biological exploratory movement and *in silico* exploratory movement, including physical and behavioral components, result from the same guiding evolutionary processes. Thus, *in silico* evolution can arrive at similar morphologies and exploratory behaviors as those found

in biological organisms. In this paper, we present preliminary results of our movement studies by evolving the morphologies and controllers of virtual creatures to successfully forage for food in a 3D physically simulated environment.

Since Sims' pioneering work (Sims, 1994b), several researches have used physical simulations of virtual creatures for evolution of locomotion (Pilat and Jacob, 2008), light-following (Pilat and Jacob, 2010), box-throwing (Chaumont et al., 2007), and co-evolutionary tasks of box-grabbing (Miconi and Channon, 2006), and fighting (Miconi, 2008). While these results provide a good basis for movement and sensing, they are not directly applicable to sustained foraging. Sustained foraging for multiple food items distributed around the environment is not demonstrated in these studies.

Evolutionary robotics approaches, e.g. (Nolfi and Floreano, 1998), are used to study artificial foraging. However, these studies are centered around the evolution of controllers of robots with fixed morphologies and deal primarily with fixed movement systems, e.g., wheeled. Evolution of sustained foraging behaviors in physical virtual creatures, where both the morphology and controller are under evolutionary control, has not been extensively studied. (Chaumont and Adami, 2011) provide one of the first examples of the evolution of sustained foraging in 3D physically simulated legged creatures albeit through a complicated experimental system with several evolutionary stages.

We present the results of experiments in the evolution of morphologies and controllers of virtual creatures foraging for a limited food resource in a virtual physical environment. Contrary to other approaches, e.g., (Chaumont and Adami, 2011), we evolve creatures through single-step evolutionary experiments in environments consisting of multiple uniformly distributed food objects. The creatures evolve successful sustained foraging ability that is resilient to changes in the number and distribution of food objects.

Virtual Creature Model

Sims' Blockies model (Sims, 1994b) is a standard model in virtual creature evolution combining a simple phenotype with a powerful generative encoding. Our model, described

The author would like to acknowledge and thank for the support of the Japan Society for the Promotion of Science (JSPS) through the JSPS Fellowship for Foreign Researchers and the JSPS Grant-in-Aid for Scientific Research.

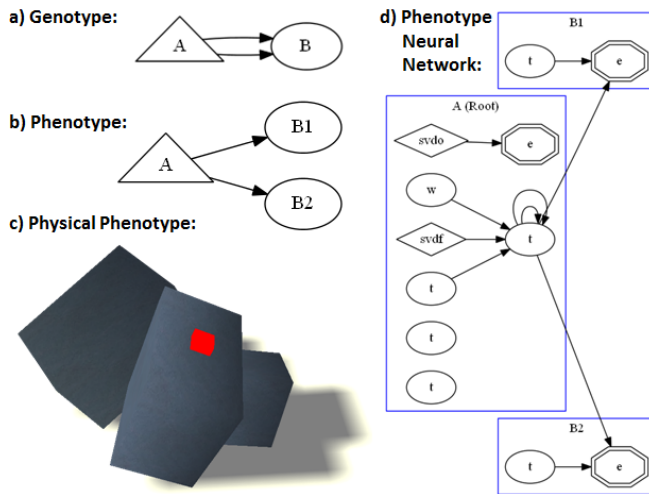


Figure 1: A sample evolved virtual creature forager showing: a) the genotype graph, b) the corresponding phenotype tree (recursion level 1), c) the corresponding physical phenotype, and d) the phenotype neural network.

in detail in (Pilat and Jacob, 2008), modifies the Blockies model by simplifying the controller model. The morphology of a virtual creature is composed of articulated cuboid body parts connected with simple hinge joints. The joints provide each body part with one degree of freedom with respect to the connected neighbor part. Simple angle limits enforce limited inter-penetration of two connected body parts. Non-connected body parts cannot inter-penetrate.

The creature phenotype is a rooted tree with nodes and edges directly corresponding to the connected body parts of the physical creature. The genotype is a directed graph with possible cycles and loops specifying a generative encoding. A recursion parameter controls the generation of the phenotype from the corresponding genotype. Fig. 1 shows the genotype and phenotype representations of a sample evolved forager. Structural parameters of the body parts and parameters controlling the building instructions of the phenotype (i.e., body part size, scaling factor, reflection, joint contact position and orientation) are stored in the genotype nodes and links. These parameters are evolved by the evolutionary system modifying the resulting creature morphology.

Virtual creatures are controlled by simple recurrent artificial neural networks that are segmented into parts embedded into body nodes as illustrated by the example in Fig. 1. Connections between neurons in the same node or neighboring nodes are allowed. In contrast to Sims' experiments, we do not use a global neural node which offers a vehicle for centralized control but slows down the evolution of simple virtual creatures evolving reactive behaviors.

Compared to the large neuron repertoire used by Sims' and derived work, e.g., (Chaumont and Adami, 2011), we

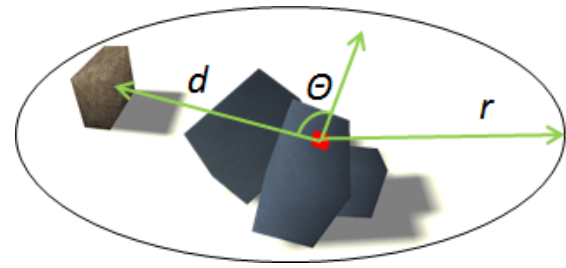


Figure 2: The spherical sensing model provides a virtual creature with information about source points (in this case, another virtual creature) in its environment within a sensing range r : the distance d to the source point and the directional angle θ to the source point.

use simple computational neurons t with sigmoidal hyperbolic tangent transfer functions. Simple sigmoidal neurons offer a functionally more realistic biological model and simplify the optimization problem solved by the genetic algorithm. Furthermore, we found that this simpler neural representation is sufficient to evolve well performing virtual creatures for various tasks. The outputs of each neuron are standardized to be in the range $[-1, 1]$. Sinusoidal periodic source neurons w are used as waveform generators that feed a periodic signal into the network.

Two types of special neurons are present for each body part: sensors and effectors. Effector neurons e power the joints of connected body parts and act as sinks of the neural network. Sensory neurons s provide the network with information gathered from the source-point vision system described in the next section. Some special neurons are not used in the phenotype of some body parts (e.g. effector neuron in main body part) but are kept in the genotype to allow the reuse of body parts during evolution.

Sensing Model

The omni-directional source-point sensing model of a virtual creature is defined by a sensing sphere of a specified radius r around the center of the creature, as shown in Fig. 2. The sensing system provides the creature with information about objects within its sensing sphere. The model is an extension to that in (Pilat and Jacob, 2010) by allowing sensing of different object classes as described below.

For each virtual creature at each simulation step, the sensing system selects the closest (by euclidean distance) source object in its sensing area for each type of sensor based on object classes. It is possible to alternate between objects in consecutive steps if the creature moves around their median point. We only allow sensing of objects belonging to other creatures of the same creature class *svds* (i.e., same population), sensing of objects belonging to other creatures of a different creature class *svdo* (i.e., different populations), and sensing of environmental food objects *svdf*.

Since each body part of a virtual creature is represented as a separate simulation object, the sensing system is able to sense body parts instead of sensing entire creatures. This is advantageous since it allows us to filter the sensed objects (e.g., only a root body part). Self-selection is not permitted. Object sensing can be filtered based on the requested object type. We can selectively enable the sensing of body parts of virtual creatures, body parts of dead virtual creatures, light objects in the environment, and other environmental objects. The creatures can also be made to sense any object irrespective of the type or no objects at all.

Once a source object is selected by the sensing system, information about its location with respect to the virtual creature is calculated and fed into the sensory neurons. This includes: distance d and angle θ , as illustrated in Fig. 2 and detailed in (Pilat and Jacob, 2010). The θ angle is the angle between the positive x -axis of the virtual creature's main body part and a vector from the center of the virtual creature to the source object. The sign of this angle specifies whether the object is positioned to the right or to the left of the virtual creature body frame direction. The distance measure d is the squared euclidean distance between the virtual creature position and the source object position, scaled to $[0, 1]$. Each sensory neuron combines the sign of the angle θ and the distance d into a single numerical value.

Simulation Environment

The experiments were performed in the Morphid Academy simulation system (Pilat and Jacob, 2008) which is a fully featured open source virtual laboratory for the evolution of functional forms called Morphids. It features physical simulation using the ODE and NVIDIA PhysX engines, graphical visualization using the OGRE engine, and a genetic algorithm based evolutionary system. In contrast to our previous work, we use the NVIDIA PhysX engine for physical creature simulation in the presented experiments. We found that the PhysX engine provides simpler control of inter-penetrations and requires less parameter optimization in order to evolve well performing virtual creatures.

The simulation environment used during evolution, called the training environment, is composed of uniformly distributed and randomly sized cuboid food objects. This differs from the experiments presented in (Pilat and Jacob, 2010) which used a single light object per evaluation. Virtual creatures are positioned randomly within the simulation area (one creature per evaluation). The random locations of both the creatures and food objects provide a different training environment each time a creature is evaluated but with the same distribution method and number of food objects. A sample foraging simulation screenshot is shown in Fig. 3.

Initially, the creatures are dropped onto the simulation surface from a specified height. Two validity checks are performed to ensure the creature movement is not due to simulation instabilities: one check while suspended over the sur-



Figure 3: Screenshot of the graphical simulation environment showing a forager and food sources. The sensor direction is indicated by a small green cube on the creature body.

face with no gravity and second check after a stabilization period on the surface. Invalid creatures are removed from the population and replaced with randomly generated ones. Creature evaluation begins after a creature passed all the validity checks and is resting on the simulation surface and continues for 50,000 or 100,000 physical simulation time steps, depending on experiment. Creatures are able to move around the environment and interact with the stationary food objects. When a creature touches a food object, the object is consumed and removed from the simulation. The sensory system of the creature is then able to load information about another closest object into the neural network.

The evolutionary system is a standard steady-state genetic algorithm using deterministic tournament selection with a tournament size $k = 3$. Each tournament evaluation is simulated independently to minimize adverse effects of sharing the simulation space. We used population sizes of 100 or 200 initialized with randomly generated virtual creatures. Genetic operators of crossover (at a rate of 20%), grafting (at a rate of 20%), and copy (at a rate of 60%) are applied to two winners of each tournament and a child individual replaces the loser in the population. Mutation is applied to the resulting child creature. The genetic operators are similar to (Sims, 1994b) and are described in (Pilat and Jacob, 2008).

Fitness is calculated using a simple consumption fitness function derived from the number of food objects consumed during the evaluation, with each food object contributing 10 fitness points. If the creature has not consumed anything, its fitness is 1. This fitness function can suffer from the bootstrap problem (Nolfi and Floreano, 1998) where the fitness during initial generations is 1 since the creatures are not able to move effectively around the environment. We experimented with alternative fitness functions that replace the fixed fitness with a movement-based fitness rewarding mov-

ing around the environment if a food object is not consumed. However, we are still able to achieve good results using the original functions and the benefits of the alternative ones are not obvious from the preliminary results.

Foraging Results

Most of our experiments evolved creatures that are able to successfully forage food objects in the environment. We analyzed foraging strategies using foraging directionality and the ability for sustained foraging. The accuracy and efficiency of the foraging behavior depends on the morphologies and movement strategies of the creatures and the testing simulation environment, as described below.

Morphologies and Controllers

The efficiency of foraging is linked to the morphologies and movement strategies of the evolved virtual creatures as discussed for locomotion tasks in (Pilat and Jacob, 2008). The creatures that evolved fixed body orientation movement using pushing or swinging movement strategies were able to evolve efficient foraging strategies. Although still successful, creatures with changing body orientations offset the movement direction slowing down the foraging time.

The size of the virtual creature (combined size of all the body parts) had an impact on the foraging strategies. Large creatures, or creatures that spread out their body parts, were more successful as they were able to sweep more food objects while moving. Compact virtual creatures had to steer directly to the food objects to touch them. More interesting behavior and better use of the limbs, akin to the often studied box-grabbing task Sims (1994a), is possible if we enforce that only the main body part can consume food.

Related to the size problem, some virtual creatures were not able to easily consume small food objects and would circle around them at first. This phenomenon seemed related to the re-orientation abilities of the virtual creatures. Some creatures, especially those with swinging strategies, could re-orient their morphologies very precisely and did not suffer from this problem. However, others that employed more complex body movement were not able to easily orient leading to missed food items and inefficient foraging.

The neural network controllers of evolved virtual creature foragers were simple with a few neurons and neural connections (example in Fig. 1). These simple neural networks provide a good example of the power of simple sigmoidal neurons as compared to the complex neural repertoire used in (Sims, 1994b) and simplify the possible fabrication of the creatures, similar to (Lipson and Pollack, 2000). Several evolved networks showed successful removal of unnecessary sensory input by eliminating connections from creature sensors while keeping the food sensor (e.g., the *svdo* neuron in Fig 1 leading into the unused effector of the root part).

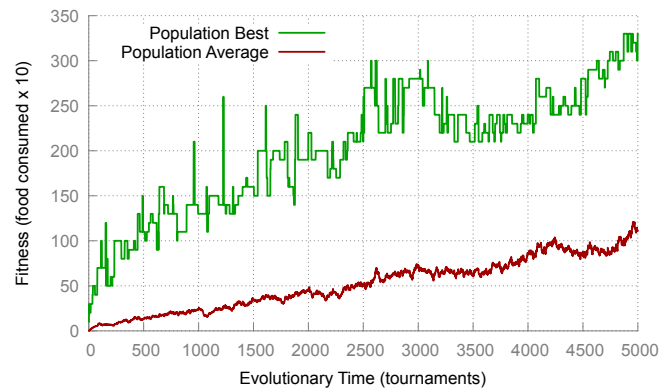


Figure 4: Fitness of an experiment with 50 food sources showing best-of-population fitness (green) and average fitness (red). Random food placement and chaotic effects of the physics engine cause the variability in best-of-population fitness of subsequent evaluations (Pilat et al., 2012).

Training Environment

The number of food objects in the training environment impacted the rate of success of evolved foraging and the accuracy of the evolved strategies. Experiments with a low number of food items (5 or less) had a difficult time evolving successful foraging strategies due to the inability of the fitness function to award movement without consumption. Experiments with 10 food items produced successful foraging strategies at a slower pace compared to the highly successful experiments using 40 or 50 items.

The distribution of food in the training environment did not impact the evolution of successful strategies. Experiments using uniformly distributed food sources and food distributed in uniformly distributed patches both produced successful foragers. The impact of the sensing range during evolution in different environments is still under investigation. Fig. 4 shows a sample fitness plot of an evolutionary run with 50 uniformly distributed food items.

Directionality of Foraging

The food object information fed to the creature neural networks contains distance and directionality components. Virtual creature controllers evolved to use the directionality information in order to orient themselves and move towards the objects. We first look at the directionality dependent foraging performance of our evolved virtual creatures.

The testing environment was composed of a single food object placed on a fixed-radius around the start position of the tested creature. The virtual creature was simulated for a fixed number of steps. The simulation was then reset, the position of the food object on the circle was changed by a fixed angle and the evaluation was repeated. With an angle of 7 degrees, we can evaluate 51 food objects on the circle.

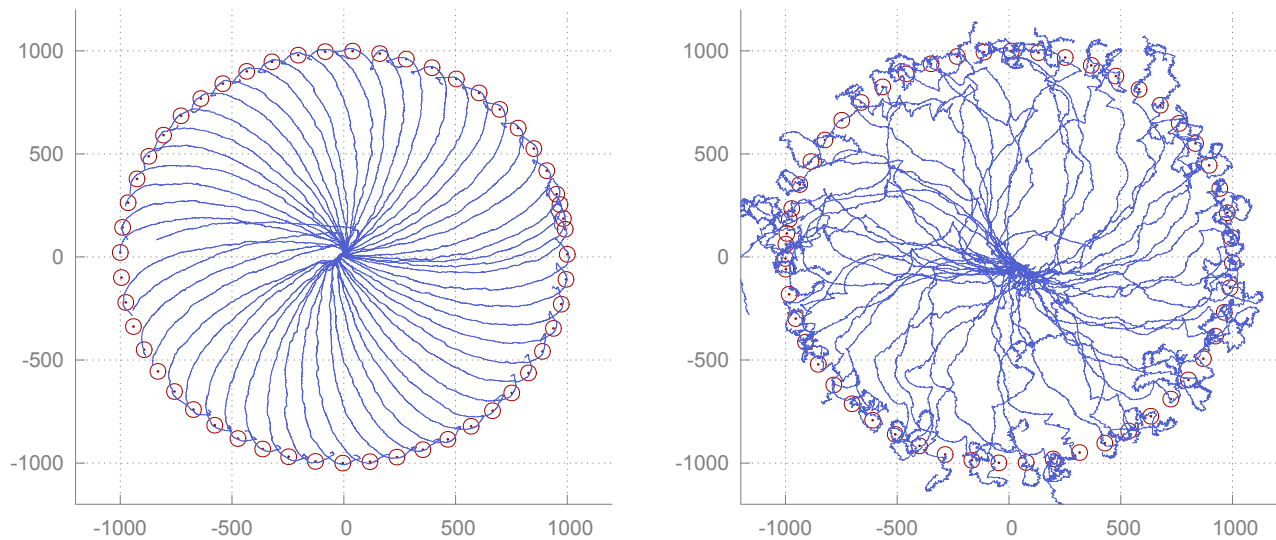


Figure 5: Composites of single-target foraging paths (blue lines) for an efficient (left) and inefficient (right) forager from a fixed start position to 51 food items (in red) spread over a circle. Each path represents a separate evaluation to a different food item.

Fig. 5 shows a composite of directional single-target foraging paths for two successfully evolved foragers. The forager on the left (from Fig. 1) can efficiently move to the food sources through slightly arched paths whereas the forager on the right uses irregular inefficient foraging paths. The foraging behavior and efficiency is highly dependent on the morphology and movement strategy of the creatures.

The initial body orientation of each evaluated creature is kept constant between evaluations. The body orientation impacts the foraging path and time as can be seen in Fig. 5. The left creature was not able to reach the leftmost food source in the given evaluation time since it was directly opposite to its body orientation - it first turned around moving in the wrong direction. A similar example is shown for the right creature looking at the irregular density of foraging paths.

Another observation that we can make from Fig. 5 deals with the movement behavior after the food source is reached. Due to the setup of the directionality evaluations, once the creature consumed the food source, it was unable to sense another one until it was reset. The creature on the right is seen to perform a random walk with no food present. The training environment provided creatures with food sources that were solely within their sensing radius. Exploratory movement when no food is sensed is an important ability of biological organisms that needs to be studied further. Evolving such behavior is critical for open-ended simulations.

A poor foraging strategy that is often seen during early stages of evolution is simple undirected movement. In a rich environment with many closely packed food sources, a virtual creature that is able to move efficiently can encounter and consume several food objects. This behavior

might be a stepping stone in evolution of successful foraging and should not be penalized during early evolution.

Sustained Foraging

Directional single food source foraging does not provide any information about foraging of multiple food sources. Sustained foraging is crucial for open-ended simulation environments where the survival and evolution of virtual creatures is related to the ability to find and consume food in the environment. To look at the sustained foraging performance, we use a testing environment with a number of uniformly distributed food items.

Fig. 6 provides multiple food foraging paths for two evolved creatures. From these paths, we can see that the creatures are able to successfully perform sustained foraging of several food objects in their environment, irregardless of the number of food objects. The efficiency of the foraging movement can be deduced from observing those plots. From the smoothness of the path in Fig. 6 (left), we are able to correctly deduce that the virtual creature can easily turn its body. Furthermore, looking at the distances between the food objects and the path, we can deduce that the creature is quite large and is able to sweep food items.

In Fig. 6 (right) we see an interesting inefficient foraging path of 50 food objects for an evolved virtual creature. This creature is not able to turn effectively in order to consume small food objects and sometimes circles around a food object, which is quite evident from its foraging path. However, it is still able to consume all of the objects in the environment, albeit with a lower efficiency.

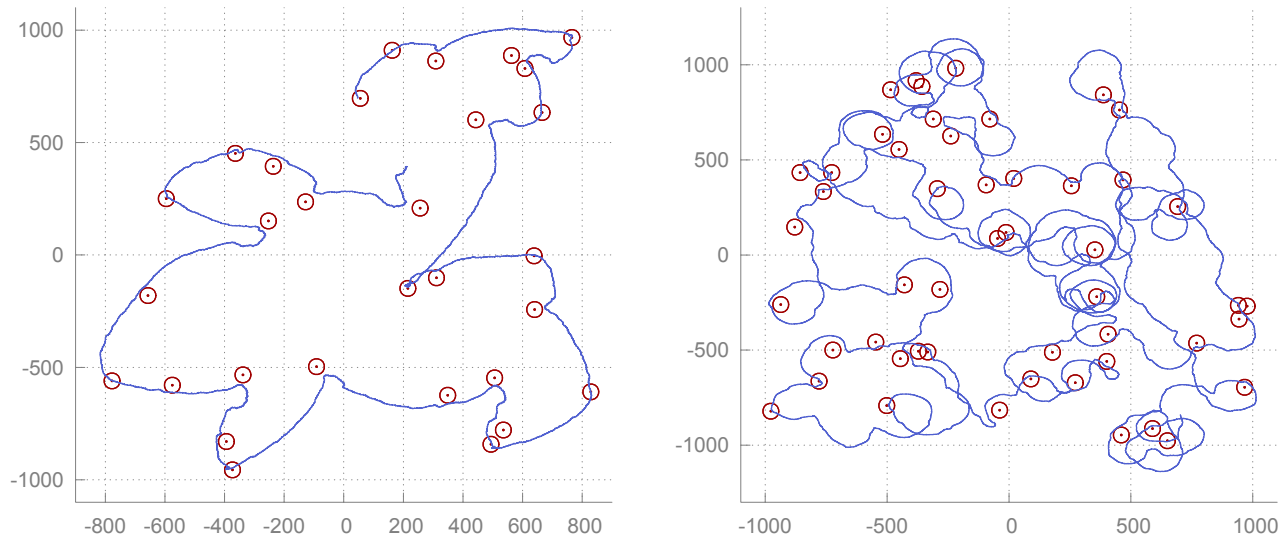


Figure 6: Sustained foraging paths (blue lines) for an efficient (left) and inefficient (right) forager from a random start position to 30 (left) and 50 (right) food items (in red). The food is uniformly distributed around the environment.

Environmental Effects

To study the resilience of the evolved foraging behaviors, we evaluated several evolved foragers in various testing environments. These testing environments differed from the training environment used during evolution in the distribution and number of food sources. Fig. 7 shows the sustained foraging paths for creatures that evolved in a random training environment evaluated using testing environments with different geometrically structured food distributions: spiral, circular, double circular, lined, and grid.

The foraging paths in Fig. 7 provide an interesting real-world application as they can approximate a Euclidean Hamiltonian path between the food items. The environment can be modified to solve the Euclidean traveling salesman problem. Although the solutions are not optimal, they can form good real-world approximations when an efficient forager is used. Since the next food object visited after one is consumed is usually the closest food object, disregarding the effects of body orienting as described above, the process is similar to the greedy nearest neighbor algorithm.

Fig. 7 also provides two examples demonstrating the tightness of the foraging path. The two concentric circle environments differ in the spacing between the circles: larger than spacing along the circle in the left environment and smaller in the right environment. This difference produces unique foraging paths for the two similar environments due to the selection of closer food objects. The grid environment spreads food objects in a regular grid pattern. The forager's choice of which equidistant food object to visit next depends on its movement and orientation just after consuming the previous object.

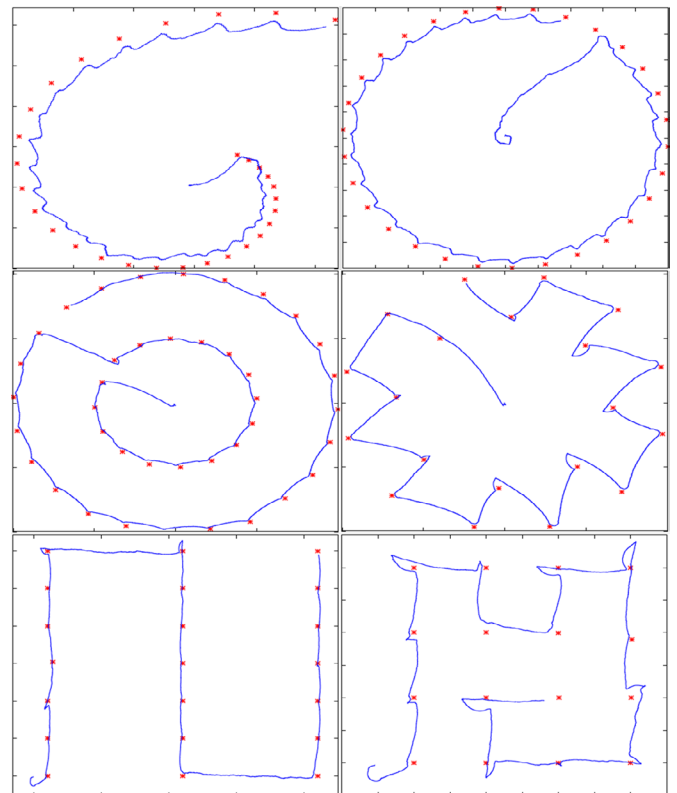


Figure 7: Foraging paths (blue lines) for several evolved foragers in different testing environments with the following geometric patterns: spiral (30 items), circular (30 items), double concentric circles (40 items), close double concentric circles (20 items), lines (21 items), and grid (9 items).

To quantitatively measure the effect of the environment on the evolved foraging behavior, we evaluated several evolved foragers in environments with different density and distribution of food objects. Density ρ was measured with Eq. 1 as the number of food objects over a square unit of simulation space. Food distribution was measured with Eq. 2 as the entropy S based on fixed partitioning of the simulation space (2D histogram estimator) into 100 simulation boxes (10 by 10). N is the number of food objects spread over a simulation area of size δ_x by δ_y and N_k is the number of food objects in the k th partition box.

$$\rho = \frac{N}{\delta_x \delta_y} \quad (1)$$

$$S = - \sum_k \frac{N_k}{N} \log_e \frac{N_k}{N} \quad (2)$$

Density evaluations varied the density by changing the size of the simulation area while maintaining an equal number of uniformly distributed food objects (50) and constant entropy (within a small variation due to the random placement). Fig. 8 illustrates the impact of the food density on foraging time for the evolved efficient forager from Fig. 5 (left). We can see that the foraging time is related to the density value with an inverse-square relationship. This is not surprising since, from Eq. 1, this produces a linear relationship between distance and foraging time.

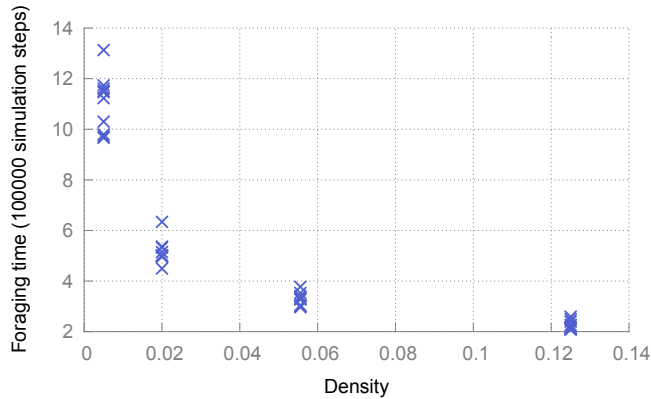


Figure 8: Foraging time to consume 50 uniformly distributed food objects with a different spread density. Each point represents a different evaluation experiment. 10 experiments were run per density value.

The entropy evaluations varied the distribution of 50 food objects in the simulation environment while maintaining a constant density value. Fig. 9 shows the entropy results of the evaluation of the evolved efficient forager in Fig. 5 (left) in environments with 7 food distributions: uniformly distributed (random), patchy with 10, 5, and 3 food patches, circular, grid, and lined. The entropy values are dependent

on the granularity of the calculation method. In our calculation, the grid arrangement filled each entropy space partition with at most 1 food object, thus maximizing the entropy equation. The patch configurations filled a low number of partitions with many objects, minimizing the entropy.

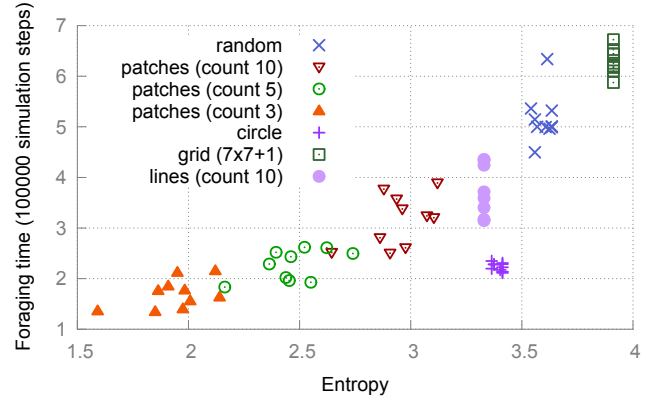


Figure 9: Foraging time to consume 50 food objects distributed using uniformly random, patchy, circular, grid, and lined distributions as indicated in the color-coded legend.

In general, the foraging time scales linearly with the entropy up to the maximum entropy value (close to the grid configuration). The variability in the 10 samples for each distribution was small for this forager. The circular distribution produced a high entropy value but with a low foraging time due to an efficient foraging path along the circle. These results indicate that the evolved forager can perform well in environments with various food distributions (varied entropy values). Evaluations with other successfully evolved foragers produced quantitatively similar results.

By evaluating several evolved foragers in the same environment, we can directly compare their foraging ability and the impacts of the shared resource foraging on foraging paths. Fig. 10 shows an example of such an evaluation using three evolved foragers and 200 food objects. We can compare the different movement strategies of each creature based on its path. The forager in orange performed worse compared to the other two foragers (due to its slow movement rate). In an example of food competition, all three foragers moved towards the last remaining food source, as seen around point $(-500, 1300)$.

Conclusions

We presented the results of experiments in evolving virtual creature foraging in physical environments containing stationary food objects. The virtual creatures were composed of articulated blocks powered by a neural network controller. The sensing system calculated and provided the neural network with distance and angle information of the position of the closest food source in the environment.

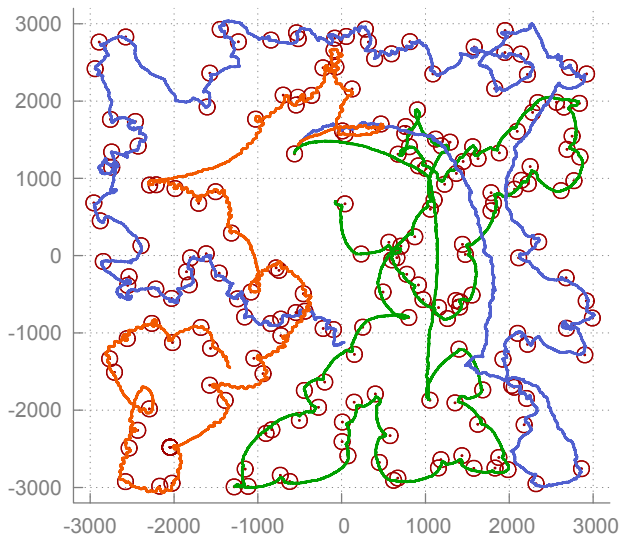


Figure 10: Foraging paths for three evolved foragers (three colored lines) consuming 200 food sources (in red) spread uniformly over the simulation environment.

The experiments successfully evolved foraging in virtual creatures of various evolved morphologies and movement strategies. The foraging behaviors were accurate with respect to food object directionality, sustainable with respect to covering multiple food sources in the environment, and resilient with respect to changes in the simulation environment. We also commented on the impact of the morphologies, movement strategies, and the training environment on the successful evolution of foraging.

These preliminary experimental results in our study of the evolution of *in silico* exploratory movement identified a range of foraging movement strategies in simple two or three body part virtual creatures. As an extension to this work, we are currently investigating the impact of the sensing range on the evolution of sustained foraging, especially in environments where the distance between food objects can be greater than the sensing range of the creatures.

This work effectively demonstrates the utility of physical simulation environments for studying biological phenomena and biological processes. The next logical step, which we are currently investigating, is to experiment with co-evolutionary settings where several virtual creatures or several populations of virtual creatures are evolved using the same physical environment. Such studies will allow us to explore the co-evolutionary dynamics of co-operative and competitive behaviors, such as the classic predator-prey scenario that is prevalent in the biological world.

The evolution of sustained foraging behaviors in physically simulated media is instrumental for future experiments in simulated open-ended environments. Foraging plays a

crucial role in such simulations and will enable virtual creatures to live, compete for food resources, and breed, thus fueling a sustainable virtual ecosystem. Evolution in this ecosystem can allow us to study speciation, group behaviors, niche construction, and other evolutionary processes that are difficult or impractical to study in natural ecosystems.

References

- Chaumont, N. and Adami, C. (2011). Evolution of sustained foraging in 3d environments with physics. *CoRR*, arXiv:1112.5116v1 [cs.NE].
- Chaumont, N., Egli, R., and Adami, C. (2007). Evolving virtual creatures and catapults. *Artificial Life*, 13:139–157.
- Holyoak, M., Casagrandi, R., Nathan, R., Revilla, E., and Spiegel, O. (2008). Trends and missing parts in the study of movement ecology. *PNAS*, 105(49):19060–19065.
- Lipson, H. and Pollack, J. B. (2000). Automatic design and manufacture of robotic lifeforms. *Nature*, 406:974–978.
- Miconi, T. (2008). Evosphere: Evolutionary dynamics in a population of fighting virtual creatures. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC2008)*, pages 3066–3073.
- Miconi, T. and Channon, A. D. (2006). Analysing co-evolution among artificial 3d creatures. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation (CEC 2006)*, pages 1639–1646. IEEE Press.
- Nathan, R., Getz, W. M., Revilla, E., Holyoak, M., Kadmon, R., Saltz, D., and Smouse, P. E. (2008). A movement ecology paradigm for unifying organismal movement research. *PNAS*, 105(49):19052–19059.
- Nolfi, S. and Floreano, D. (1998). Co-evolving predator and prey robots: do arms races arise in artificial evolution? *Artificial Life*, 4:311–335.
- Pilat, M. L. and Jacob, C. (2008). Creature academy: A system for virtual creature evolution. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC2008)*, pages 3289–3297. IEEE.
- Pilat, M. L. and Jacob, C. (2010). Evolution of vision capabilities in embodied virtual creatures. In *Proceedings of the 12th annual Conference on Genetic and Evolutionary Computation (GECCO2010)*, pages 95–102. ACM.
- Pilat, M. L., Suzuki, R., and Arita, T. (2012). Dealing with rounding error problems in evolutionary physical simulation. *Artificial Life and Robotics*, 17 (in press).
- Sims, K. (1994a). Evolving 3d morphology and behavior by competition. In *Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems (Artificial Life IV)*, pages 353–372.
- Sims, K. (1994b). Evolving virtual creatures. In *21st International ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH94)*, pages 15–22. ACM.
- Wake, M. H. (2001). Bodies and body plans and how they came to be. In Kress, W. and Barrett, G. W., editors, *A New Century of Biology*, pages 29–52. Smithsonian Books.

Deformable Octahedron Burrowing Robot

Juan Cristobal Zagal¹, Cristobal Armstrong¹ and Shuguang Li²

¹Department of Mechanical Engineering, University of Chile, Santiago, Chile

²School of Astronautics, Northwestern Polytechnical University, Xi'an, Shaanxi, P.R. China
jczagal@ing.uchile.cl

Abstract

This paper explores the use of a deformable octahedron robot as an alternative for the autonomous exploration of complex confined spaces, voids and tunneling structures. Current robotic platforms lack the capabilities for adapting their shape when moving through intricate sections of cavities. We discuss the geometrical and dynamical properties of a deformable octahedral platform. We use real and simulated robots to test and synthesize locomotion controllers that allow our robots to travel along different portions of a tunneling test bed. Evolutionary methods allow us to automatically produce controllers for in-pipe motion. We demonstrate the capabilities of peristaltic locomotion, different modes of deformation and volumetric adaptation. An evaluation of motion capabilities inside pipe elbows and branches is performed. Our results suggest that this type of deformable robot has a potential for travelling along confined spaces.

Introduction

Current robots have limited capabilities to access confined spaces such as narrow caves, complex pipeline networks, bifurcating blood vessels and uncharted pipeline networks. Various machines have been proposed for autonomous navigation under confinement. In-pipe robots include wheeled, caterpillar, wall-press, inchworm, screw, walking and even snake-like [19] devices. Flexible catheters have been also developed for robotically assisted surgery [8] as well as many caterpillar-like platforms for disaster and mine exploration [5].

Although robotics snakes and catheters can curve, most devices lack of the capability to deform and shift their shape adapting to the various geometries that might arise under confinement. Deformation seems to be an important capability to be further developed for the autonomous exploration of confined spaces that arise in mines, the human body, collapsed buildings, industrial and marine pipelines, etc.

Although legged animals are successful at travelling over relatively flat terrain (horse, cheetah, etc.), soft deformable invertebrates such as worms, slugs and leeches are the masters of confinement. Earthworms are able to travel underground by exploiting waves of muscular contractions that alternatively shorten and lengthen different portions of their body. Since the shortened part also widens, it can be anchored to the surrounding soil, allowing the narrowed lengthened part to move forward, following a peristaltic pattern [16].

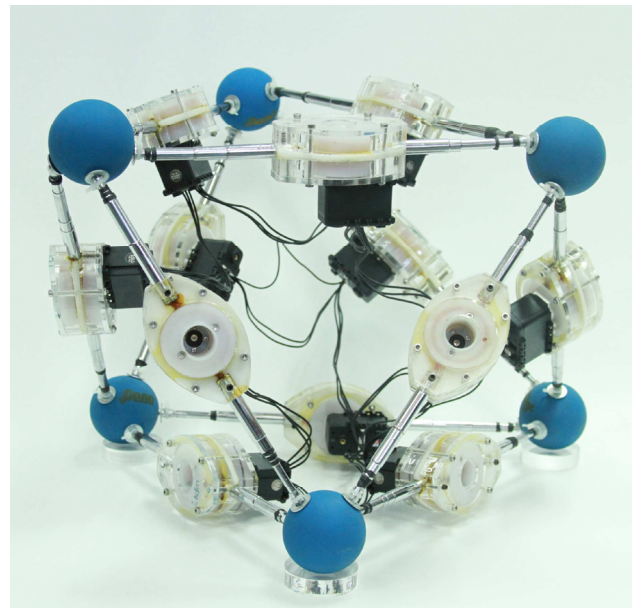


Figure 1: The octahedron burrowing robot. Edges are composed by linear actuators and the vertices are covered by rubber balls used as anchoring material.

The recently expanding literature on deformable robotics illustrates interesting developments on materials, methods, path planning and locomotion of deformable robots, usually on flat terrain. Less attention has been devoted to exploring the applications of these concepts to the exploration of confined environments.

Rather than exploring rugged planar terrain, deformable robots might have a great potential for traveling inside voids and confined spaces. In this study the goal is to explore the capabilities of a deformable octahedron robot to penetrate, travel and transition between cavities and tunneling structures.

Our first prototype uses hydraulic linear actuators and the second was constructed with motorized electric linear actuators, constructed by spinning a drum loaded with a plastic line, following the same principle of power car antennas.

The remainder of this paper is organized as follows: The next section introduces related work, and then the octahedron platform is introduced together with its simulation. Force

feedback is analyzed as well as studies of locomotion inside various pipeline joints. Finally the conclusions are presented.

Related Work

Deformable robots

The ability to significantly deform, adapt and expand, at a much higher level than conventional robotics, enables soft robots to access environments today restricted to conventional autonomous machines. It appears that nature has provided soft animals with extraordinary abilities to control their body even with few muscles and little dedicated neural circuitry [17].

Complex, yet coordinated, motor interactions are apparently obtained from the dynamical coupling between locally regulated muscular structures, enabling organisms to perform control tasks that would otherwise be attributed to centralized neural computation. This ability is related to the recently conceptualized terms of morphological communication [18] and morphological computation [15], where the mechanism itself is used as a form of mind.

The remarkable capabilities of crawling and jumping by a mostly circular soft robot were recently demonstrated in [22]. Active continuous deformation allowed the structure to locomote over rough terrain. Deformation was achieved by extending or shrinking eight shape memory alloy (SMA) coils distributed along the circular perimeter. Due to the high driving voltage and power required by these actuators, the device was tethered during experiments.

Peristaltic locomotion in soft robotics has been demonstrated with different materials. A flexible braided mesh-tube was wrapped with a network of antagonistic NiTi coil actuators in [21]. The prototype, inspired by the hydrostatic skeleton of the *Oligochaeta* worm, demonstrated robust tethered locomotion over a planar horizontal surface. Interestingly, locomotion persisted after impacts with a hammer were applied to the mechanism.

Soft pneumatic actuators were used in [3], demonstrating how selective inflation of multiple cells along a worm-like body can generate peristaltic motion. The soft cells were constructed using two layers of silicone: a flat layer embedding a fabric mesh, and a thicker expandable layer that produces bimorph bending of the compound when inflated.

The same selective inflation principle was applied by arranging the cells as a circular belt. The resulting ring was able to roll autonomously on a flat surface. Similarly, a peristaltic pattern of motion was also achieved by combining three pneumatic McKibben actuators in series [11]. Their prototype for an autonomous peristaltic endoscope was tested inside a horizontal tube with slight curvature and slope.

Snake-like robots that exhibit peristaltic locomotion have been analyzed in studies like [12,20]. Forward locomotion capabilities are usually studied but less attention has been paid to exploring rotation and volumetric adaptation to the various geometries that might arise in cavities. Turning patterns over the plane were analyzed for the case of a peristaltic robot studied in [13].

Tensegrity and Lattice Robots

Deformable tensegrity robots are composed of an actuated group of struts and cables. A network of struts under pure compression is supported by a continuous network of cables under tension, defining a stable volume in space. Structural morphing is achieved when varying the length of cables or struts. The shape-shifting capabilities of tensegrities enable them to locomote [7].

Tensegrities are highly deployable structures, capable of occupying a large volume when extended or a small space when contracted. They can bend themselves while their constituent elements do not experience any bending torque, since they are only subject to axial forces. When subjected to stress, the structural members are unidirectionally loaded, without reversals in the direction of member load [23]. These properties allow the simplification of element design and control.

The design and control of planar tensegrity models was studied in [6]. Controllers were generated to achieve robust performance and stabilization in the context of manipulation. Design methodologies were given to meet dynamical stiffness and vibration isolation specifications. The design and control for locomotion of more complex tridimensional tensegrities was studied in [14].

Tetrahedral robots are another form of lattice-based deformable robotics, which have mainly been explored for aerospace applications. In [4], the space-filling properties of tetrahedral robots are highlighted as an alternative for mobility on irregular terrain. However, locomotion experiments reported with this type of robot are restricted to planar surfaces [1,2]. They have demonstrated locomotion by tumbling tetrahedra over irregular, but mostly planar, terrain. It is suggested that these robots also have good capabilities for traveling over terrain with high slopes and varying obstacle sizes.

Odin is a great example of a deformable lattice modular robot specification [10]. Rather than defining a particular configuration, Odin defines a set of modules (joint, telescopic actuator and passive rod) that can be used for the construction of arbitrary deformable lattice geometries. Some experiments are reported on basic motion capabilities of a robot constructed using such modular specification. Unfortunately, the robot is hardly reproducible due to the high module cost.

Octahedron Robot

Octahedron Geometry

An octahedron is a polyhedron having eight faces. A regular octahedron belongs to the Platonic solids family. It is made by eight equilateral triangles; four triangles meet at each one of its six vertices. An octahedron has 12 edges. Figure 2 shows an illustration of a planar deployment of a regular octahedron together with different 3D views of the same solid geometry. A deformable octahedron has very interesting space filling properties which enable it to be an excellent platform for the exploration of cavities.

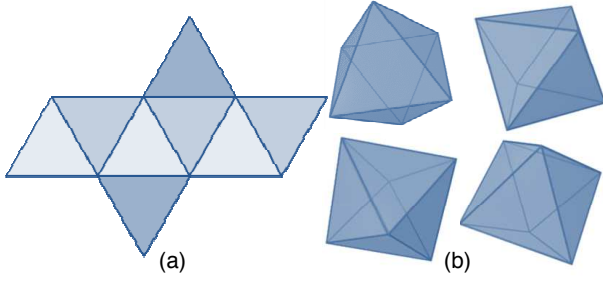


Figure 2: a, planar deployment of a regular octahedron. b, different views of the same regular octahedron.

Simulation Model

We implemented a physical simulation of the octahedron robot using the Open Dynamics Engine (ODE). The simulation contains twelve linear actuators serving as the edges of a polyhedron. Four linear actuators meet at each vertex having ball type ODE joints as motion constraints. PID dynamic compensators were used to control the force applied to each actuator while following an actuator length reference signal. The simulation allows investigating shape shifting and locomotion caused when varying the lengths of the platform edges. Figure 3 shows our simulation model under the canonical equilateral configuration.

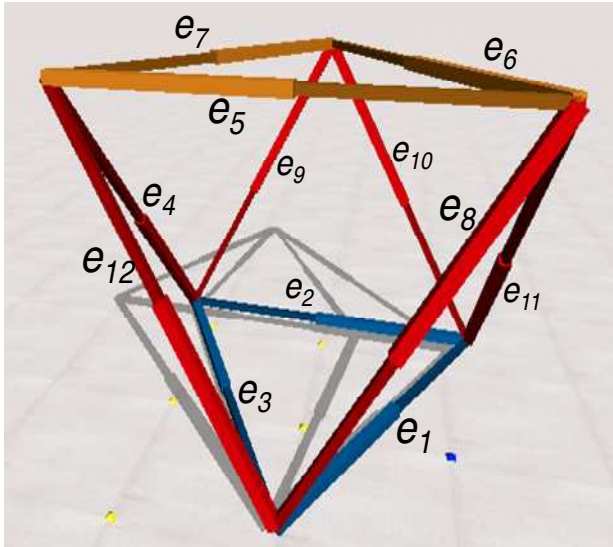


Figure 3: Simulation model implemented using the Open Dynamics Engine. The twelve linear actuators are shown at the edges $\{e_1, \dots, e_{12}\}$ of the platform.

Analysis of Deformation Modes

To begin studying the locomotion capabilities of the octahedron robot, we analyze different deformations by

looking at the amount of power required by linear actuators to sustain different configurations at steady state. The octahedron is a highly redundant over-actuated system, and it is natural to expect that some motor commands will over stress the structure, due to antagonistic force patterns that propagate along the structure. Furthermore, it is important to identify the group of natural deformations that require a minimum amount of sustaining power, allowing for graceful motion.

The understanding of the force requirements of different deformation modes will enable the promotion of natural modes during locomotion, as well as the avoidance of antagonistic modes that can eventually harm the structure and drain excessive power. In addition, a good understanding of the force patterns that arise due to intrinsic actuation, might serve to identify patterns that can be only explained by interaction with the environment.

We note a commanded deformation by a row vector c (eq. (1)) of twelve target reference positions $\{r_1, \dots, r_{12}\}$ for the linear actuators $\{e_1, \dots, e_{12}\}$ shown in Figure 3, so that r_i is the reference position for the actuator at edge e_i . Another convenient representation is a 4×3 matrix C that groups motion relevant segments in rows (eq. (2)). This notation allows identifying the symmetries exploited by different deformations.

$$c = [r_1 \ r_2 \ r_3 \ r_4 \ r_5 \ r_6 \ r_7 \ r_8 \ r_9 \ r_{10} \ r_{11} \ r_{12}] \quad (1)$$

$$C = \begin{bmatrix} r_5 & r_6 & r_7 \\ r_{11} & r_{10} & r_9 \\ r_4 & r_{12} & r_8 \\ r_1 & r_2 & r_3 \end{bmatrix} \quad (2)$$

Some natural deformation modes, that can be intuitively derived, are shown in Figure 4. The canonical configuration is shown in Figure 4a. We can describe this mode by

$$c_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \quad (3)$$

Global expansion and contraction (Fig. 4b, c) of the platform might be important for adapting to the different sizes of a given cavity. This mode can be represented by

$$c_1 = \alpha \cdot [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \quad (4)$$

Where α is the scaling constant that modulates the deformation. Expansion of a single face allows anchoring on the cavity surface with just three edges (Figure 4d, e). Examples of face expansion modes are:

$$c_2 = \alpha \cdot [1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \quad (5)$$

$$c_3 = \alpha \cdot [0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0] \quad (6)$$

Relative rotation of parallel faces (Figure 4f) allows further adaptation of the platform to the cavity internal geometry.

$$c_4 = \alpha \cdot [0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0] \quad (7)$$

Extension of the robot orthogonal to the anchoring faces is another fundamental mode of locomotion (Figure 4g) since it allows transitioning from different anchoring points, corresponding to the extension phase of peristaltic motion.

$$c_5 = \alpha \cdot [0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1] \quad (8)$$

Rotation of one face place with respect to its counter face is another natural mode of motion that might be used for accessing branches of a tunneling structure or cavity.

$$c_6 = \alpha \cdot [0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0] \quad (9)$$

Continuing this analysis might lead to the identification of various other natural modes that can be useful for locomotion of the structure. However, we would like to discover and characterize automatically the different motion modes of the octahedron structure. We present in the next section a method for the characterization of motion modes for highly over actuated structures.

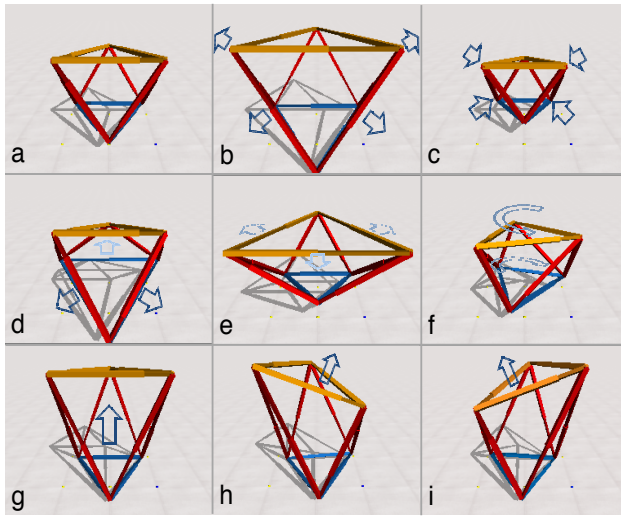


Figure 4: Example of some natural modes of deformation of the octahedral platform. a, Canonical configuration. b, Equilateral expansion. c, Equilateral contraction. d, Expansion of base face. e, Expansion of top face. f, Rotation relative to base and top faces. g, Face relative extension. h, Rotation of one counter face relative to the other. i, Same as in h but in another direction.

Automatic Characterization of Deformation Modes

Force feedback signals resulting from linear actuators can be used to sense characteristics of the surrounding environment touched by the robot (geometry, roughness, stiffness, etc.).

Sensing the intrinsic distribution of forces might be useful for locomotion and shape shifting. The interpretation of force patterns might serve to minimize global energy consumption, preserve adequate levels of stress along the structure, and even sense structural damage.

Due to the complexity of the robot, motor commands might produce over stress and even harm the structure. Eventually a controller might incorporate force feedback as a means for smooth locomotion of the machine along a cavity or pipe.

The physical simulation of the octahedron robot allows us to rapidly test force distributions resulting from any commanded deformation. We have decided to exploit this advantage by testing a large set of deformations. This set is defined by all possible deformations that can be obtained by expanding or not, by a small amount $\alpha = \varepsilon$, each linear actuator. This results in a total of $|c| = 2^{12} = 4096$ possible deformations to be evaluated.

We analyzed each deformation starting with the robot under the canonical configuration c_0 , then, we commanded a deformation vector c_T at time $t = 0$, and we computed the total power applied by the linear actuators at evaluation time $t = 100s$. The idea was to check the amount of power required to sustain c_T during steady state. Since each linear actuator carries its own PID dynamic compensator, the amount of power is proportional to the overall force resulting over the edge as a consequence of the intrinsic actuation.

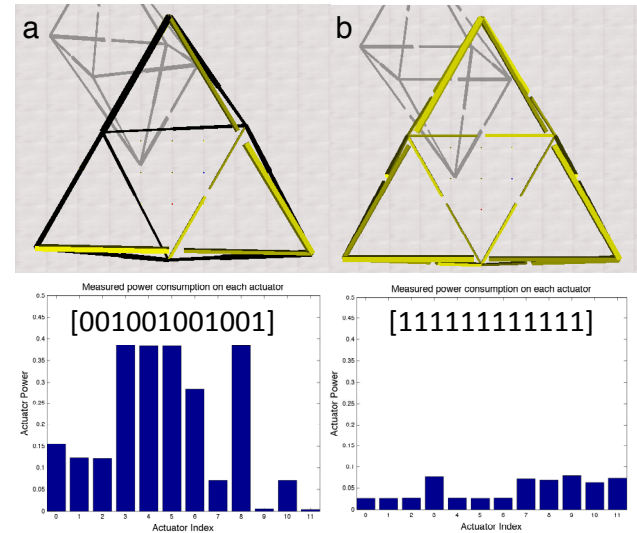


Figure 5: Power consumption measured on each linear actuator. Edges having a positive target (expanding) are shown in yellow. Those with extension target equal to zero are shown in black. a, A non-natural deformation. b, Equilateral expansion.

To ensure that forces are only intrinsic, due to the internal compensation required to sustain the target deformation, as well as to the properties of the octahedron geometry, we lifted the octahedron from ground and we set gravity to zero. Figure 6 shows plots of the total power that resulted for each deformation tested. Results appear sorted in ascending (a) and descending (b) order of total power.

These results are interesting, since they allow identifying groups of deformations having similar power consumptions. Moreover, one can easily note three groups of data, namely the natural deformations characterized by low energy consumption (blue ellipse), an intermediate group of deformations (gray ellipse), and a set of deformations characterized by high power consumption (red ellipse).

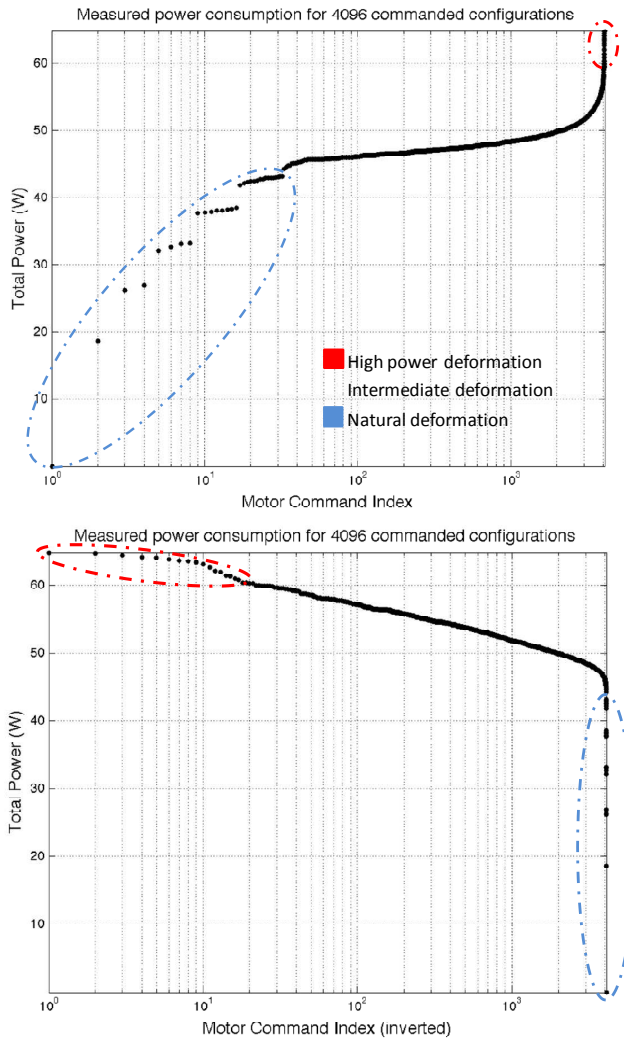


Figure 6: Total power consumption required by each commanded deformation under analysis. Results are shown in ascending order (top) and descending order (bottom) of total power consumption. A proposed distinction between high power, intermediate and natural deformation is indicated with dashed ellipses.

Evolving Basic Locomotion Modes

Several control strategies can be applied for commanding such a redundant, over actuated platform. A model based approach would require a geometrical representation describing the space of deformations that preserve the structure. The method described in previous section is a step

toward obtaining such representation. Shape shifting under the above mentioned natural modes of deformation is in general consistent with the remainder of the structure and therefore it requires small amounts of energy and force.

We first studied locomotion inside a simulated pipeline. A peristaltic locomotion controller was intuitively defined by six phases of motion, corresponding to: (1) expansion of the front face c_3 , (2) contraction of rear face c_2 , (3) contraction of c_5 edges, (4) expansion of rear face c_2 , (5) contraction of front face c_3 , and (6) extension of c_5 edges.

We defined a space of controller solutions with the parameters of maximum edge extension l_{max} , and duration of each motion phase $\{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6\}$. We used a simple genetic algorithm to search the space of possible solutions. A genome was represented by the vector $g = \{l_{max}, \tau_{3-6}, \tau_1, \tau_2, \tau_4, \tau_5\}$. To enforce peristaltic symmetry, we used the same duration (τ_{3-6}) for the phases of contraction and expansion of c_5 edges.

Crossover was performed with a probability $P_c = 0.8$ and mutation with a probability $P_m = 0.01$. After nearly 100 generations we obtained a $\sim 40\%$ of speed increase with respect to the starting engineered solution. The population size was set to 20 individuals. Figure 7 shows different stages of vertical locomotion inside a simulated straight pipe.

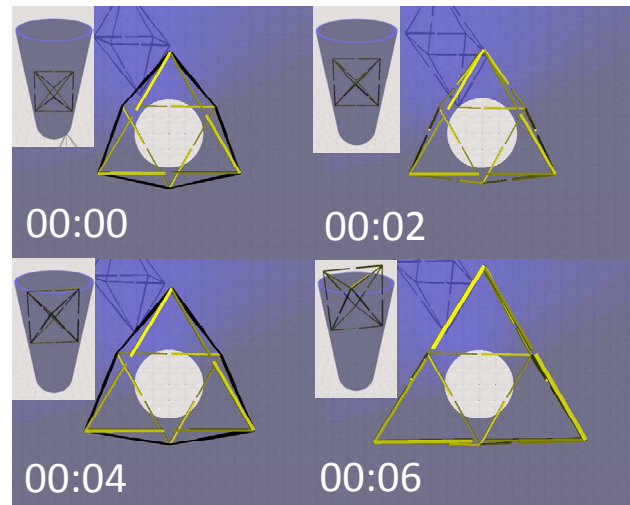


Figure 7: Simulation of octahedron platform traveling inside a vertical pipe. The robot is able to climb up the pipe interior while executing the peristaltic controller.

Navigation along {L,T,Y}-shaped pipelines

Many in-pipe robots are able to navigate along horizontal straight pipelines [19]. However, some pipeline configurations are particularly challenging for these machines. This is the case of pipeline branches and elbows.

A main problem is due to the internal geometrical changes that a robot faces when moving along these structured cavities. Figure 8 shows the group of nine pipe-joints that we have selected to test the motion capabilities of the octahedron platform.

We have considered L-shaped, T-shaped and Y-shaped joints of varying degrees of smoothness. The joint smoothness

increases toward the right hand side of the figure. A dashed arrow on the left indicates the target robot trajectory on each pipe-joint.

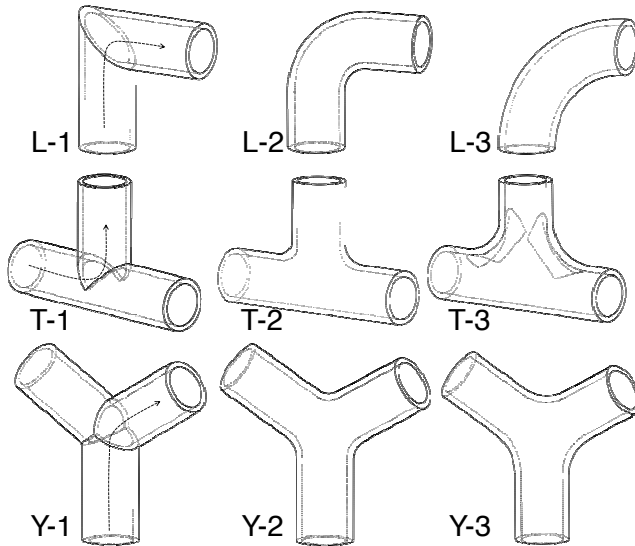


Figure 8: Pipeline elbows and branches used for testing the motion capability of the octahedron robot. We used L-shaped, T-shaped and Y-shaped pipeline joints with three different degrees of smoothness (increasing toward the right). The target robot path is shown with a dashed arrow on the left.

We carried out ten simulation trials per joint. During each trial, the experimenter was able to switch the orientation of peristaltic motion to be either lateral or vertical. This was particularly useful for motion inside T-shaped joints.

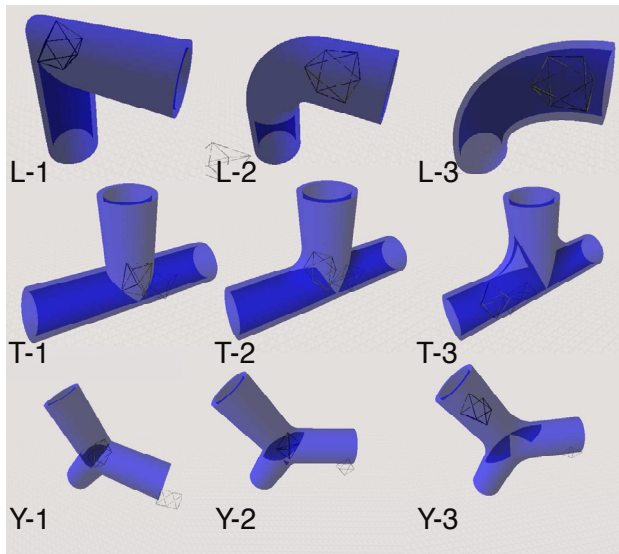


Figure 9: Screenshots taken during each joint simulation. The octahedron is shown while following the target path indicated in previous figure.

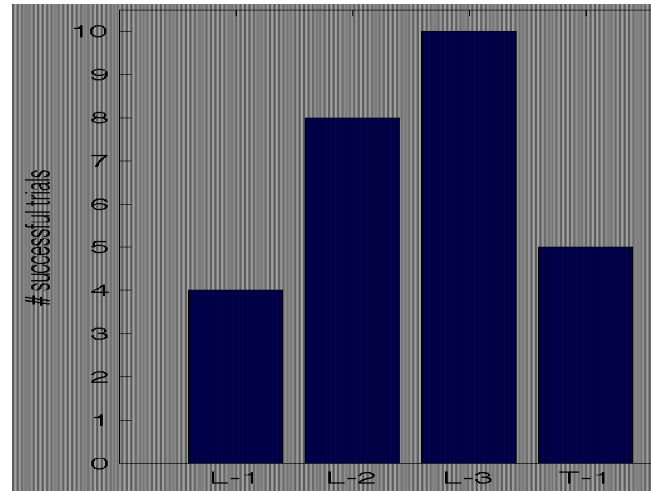


Figure 10: Number of successful trials out of a 10-trial test run. Results are shown for each joint under analysis.

A trial was counted as successful if the experimenter was able to drive the robot along the corresponding target path within a limited period of 15 s. Figure 9 shows screenshots of motion evaluation trials. The resulting number of successful trials per joint is shown in Figure 10.

The robot was able to travel along the target path one every joint. The resulting number of successful trials is not yet a statistically relevant indicator, but it allows us to identify L-3 as the joint that can be most easily surpassed. The joints L-1, T-2, T-3 and Y-1 appear as the most difficult to surpass.

Real Robots

We implemented the octahedron robotic concept with two real prototypes; the first is a hydraulic robot that uses syringes as linear actuators. A board of syringes is used for manual actuation. This robot is presented in Figure 11. We also performed locomotion experiments which are shown in Figure 12. The device was able to move at nearly one meter per minute when manually actuated.

We also built an electrically actuated robot which is shown in Figure 1, at the beginning of this paper. Both devices are tethered. The operation of the hydraulic device was aided by the force feedback transmitted along the water filled lines. We are currently working toward obtaining force feedback signals from the electrically actuated robot.

Figure 13 shows design details of the implemented electric linear actuators. A longitudinal cut of the actuator is presented together with an exploded view showing the different components.

It is important to mention that the construction was possible thanks to the use of a laser cutter. Some parts were machined using classical methods, although they can be easily fabricated with 3D printing. Figure 14 shows shape shifting tests performed with the electric prototype.

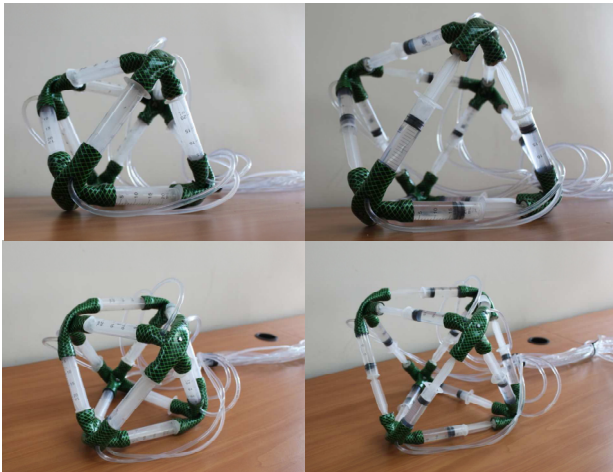


Figure 11: Hydraulic prototype constructed using syringes. The prototype is remotely actuated manually.

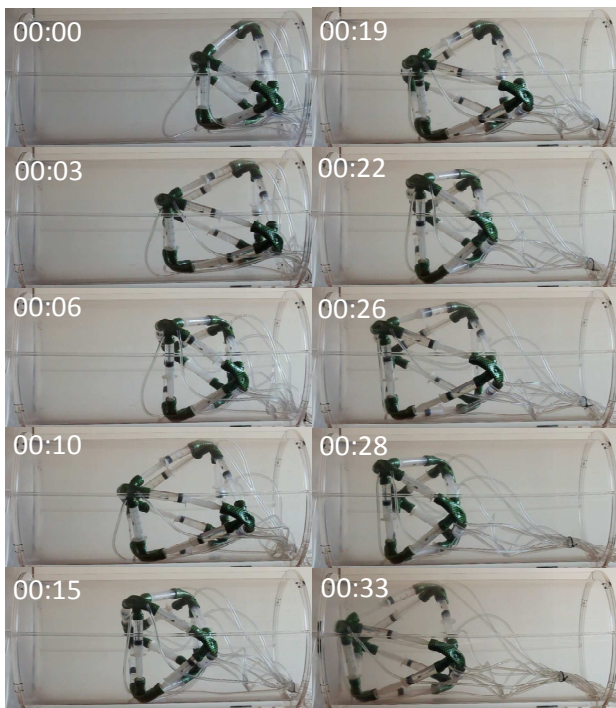


Figure 12: Locomotion inside a real pipe, ~60 cm length. The peristaltic controller was applied on a real setup showing successful lateral motion.

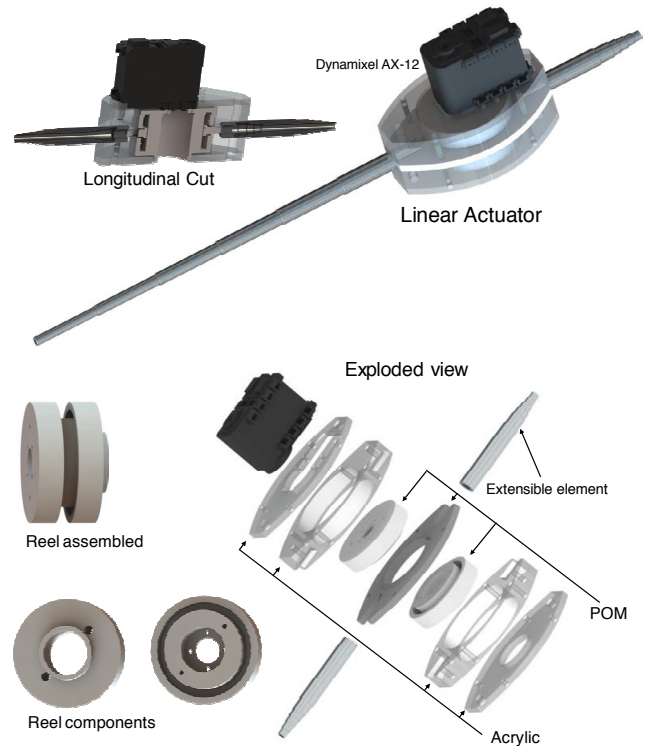


Figure 13: Different views of the design and components used for the construction of the electric linear actuators. The robot can be easily reproduced using digital fabrication techniques, such as laser cutting and 3D printing.

Conclusions

We have shown how an octahedron robot is able to travel under confinement. The robot is able to navigate along different simulated {L,T,Y}-shaped pipe joints. We have evolved a motion controller for the lateral displacement of this new robotic platform. In addition, we have presented a method to automatically explore and characterize structural deformations in terms of energy consumption. Using this method we have detected three groups of deformations which are defined by either low (natural), intermediate and high power demands. Eventually, a sense of touch might be derived from a thorough understanding of force feedback signals of the octahedral structure.

Acknowledgements

This research was funded by Fondecyt project number 11110353. We thank the thorough and detailed revision provided by anonymous reviewers.

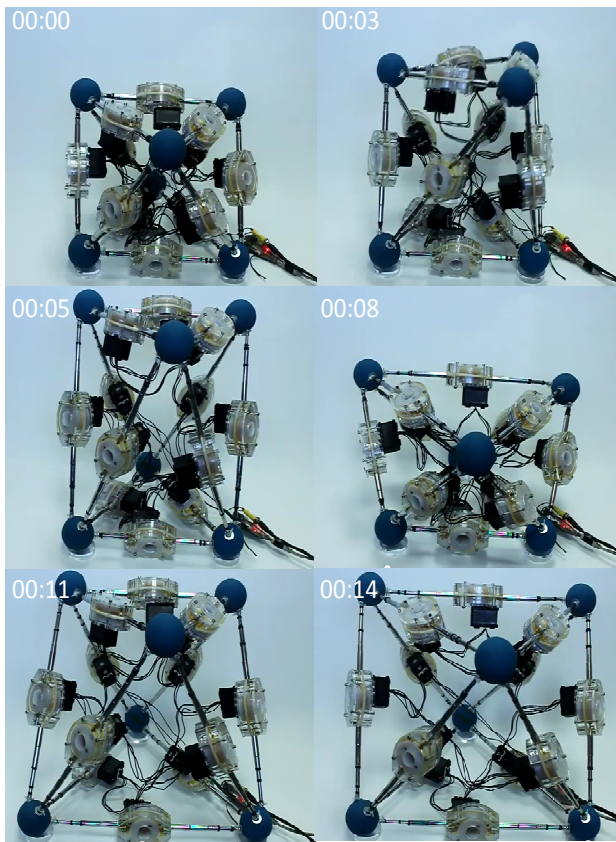


Figure 14: Illustration of shape shifting capabilities of the electric prototype. Natural modes of deformation were tested.

References

1. Abrahantes, M., Silver, A. and Wendt, L. (2007). Gait design and modeling of a 12-tetrahedron walker robot. In *Thirty-Ninth Southeastern Symposium on System Theory*, pages 21–25. IEEE, Macon, GA.
2. Abrahantes, M., Nelson, L. and Doorn, P. (2010). Modeling and Gait Design of a 6-Tetrahedron Walker Robot. In *42nd Southeastern Symposium on System Theory*, pages 248–252. IEEE, Tyler, TX.
3. Correll, N., Onal, C. D., Liang, H., Schoenfeld, E., & Rus, D. (2010). Soft Autonomous Materials — Using Active Elasticity and Embedded Distributed Computation. *12th International Symposium on Experimental Robotics*. New Delhi, India.
4. Curtis, S., Brandt, M., Bowers, G., Brown, G., Cheung, C., Cooperider, C., Desch, M., et al. (2007). Tetrahedral Robotics for Space Exploration. *IEEE Aerospace and Electronic Systems Magazine*, 22(June), 22–30.
5. Davis, A. (2002). Urban search and rescue robots: from tragedy to technology. *IEEE Intelligent Systems*, 17(2):81–83.
6. de Jager, B., & Skelton, R. E. (2005). Input-output selection for planar tensegrity models. *Control Systems Technology, IEEE Transactions on*, 13(5), 778–785.
7. Fest, E., Shea, K., & Smith, I. (2004). Active tensegrity structure. *Journal of Structural Engineering*, 130(October), 1454–1465.
8. Gayle, R., Lin, M.C. and Manocha, D. (2005). Constraint-based motion and planning of deformable robots. In *Proceedings of the 2005 IEEE International Conference Robotics and Automation*, pages 1046–1053. Barcelona, Spain.
9. Gayle, R. and Segars, P. and Lin, M.C. and Manocha, D. (2005b). Path planning for deformable robots in complex environments. In *Proceedings of the Robotics Science and Systems Conference (RSS)*, pages 225–232. Cambridge, MA.
10. Lyder, A., Garcia, R., & Stoy, K. (2008). Mechanical design of Odin, an extendable heterogeneous deformable modular robot. *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on* (pp. 883–888).
11. Mangan, E. V., Kingsley, D. A., Quinn, R. D., & Chiel, H. J. (2002). Development of a peristaltic endoscope. *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on (Vol. 1, pp. 347–352). IEEE*.
12. Nakamura, T., Kato, T., & Iwanaga, T. (2006). Development of a Peristaltic Crawling Robot Based on Earthworm Locomotion. *Journal of Robotics and Mechatronics*, 18(3), 299–302.
13. Nakamura, T. (2008). Locomotion and turning patterns of a peristaltic crawling earthworm robot composed of flexible units. 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 1630–1635).
14. Paul, C. and Valero-Cuevas, F.J. and Lipson, H. (2006). Design and control of tensegrity robots for locomotion. *IEEE Transactions on Robotics*, 22(5):944–957.
15. Pfeifer, R., Iida, F. (2005). Morphological computation: Connecting body, brain and environment. *Japanese Scientific Monthly*, 58(2), 48–54.
16. Quillin, K. (1999). Kinematic scaling of locomotion by hydrostatic animals: ontogeny of peristaltic crawling by the earthworm *lumbricus terrestris*. *Journal of Experimental Biology*, 202 (Pt 6)(6), 661–74.
17. Rieffel, J., Trimmer, B., & Lipson, H. (2008). Mechanism as Mind: What Tensegrities and Caterpillars Can Teach Us about Soft Robotics. *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems* (pp. 506–512). Winchester, U.K.: MIT Press, Cambridge, MA.
18. Rieffel, J.A. and Valero-Cuevas, F.J. and Lipson, H. (2010). Morphological communication: exploiting coupled dynamics in a complex mechanical structure to achieve locomotion. *Journal of the Royal Society Interface*, 45(7):1742–5689.
19. Roh, S. and Choi, H.R. (2005). Differential-drive in-pipe robot for moving inside urban gas pipelines. *IEEE Transactions on Robotics*, 21(1):1552–3098.
20. Saga, N., & Nakamura, T. (2004). Development of a peristaltic crawling robot using magnetic fluid on the basis of the locomotion mechanism of the earthworm. *Smart Materials and Structures*, 13(3), 566–569.
21. Seok, S., Onal, C. D., Wood, R., Rus, D., & Sangbae, K. (2010). Peristaltic locomotion with antagonistic actuators in soft robotics. *IEEE International Conference on Robotics and Automation (ICRA)* (Vol. 60).
22. Sugiyama, Y. (2004). Crawling and jumping of deformable soft robot. *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3276–3281). Sendai, Japan.
23. Skelton, R. (2001). Dynamics of the shell class of tensegrity structures. *Journal of the Franklin Institute*, 338(2–3), 255–320.
24. Trimmer, B.A. and Takesian, A.E. and Sweet, B.M. and Rogers, C.B. and Hake, D.C. and Rogers, D.J. (2006). Caterpillar locomotion: A new model for soft-bodied climbing and burrowing robots. In *Proceedings of the 7th International Symposium on Technology and the Mine Problem*, Monterey, CA.

Synthetic Biology

Papers

Modeling Scalable Pattern Generation in DNA Reaction Networks

Peter B. Allen, Xi Chen, Zack B. Simpson and Andrew D. Ellington*

University of Texas at Austin
Institute for Cell and Molecular Biology
andy.ellington@mail.utexas.edu

Abstract

We have developed a theoretical framework for developing patterns in multiple dimensions using controllable diffusion and designed reactions implemented in DNA. This includes so-called strand displacement reactions in which one single-stranded DNA hybridizes to a hemi-duplex DNA and displaces another single stranded DNA, reversibly or irreversibly. These reactions can be designed to proceed with designed rate and molecular specificity. By also controlling diffusion by partial complementarity to a stationary, cross-linked DNA, we can generate predictable patterns. We demonstrate this with several simulations showing deterministic, arbitrary shapes in space.

Introduction

Pattern formation is biologically and technologically important. Biomimetic methods for moving from top-down to bottom-up formation of designed patterns and materials have the potential to revolutionize manufacturing by dramatically reducing costs. These approaches include biomimetic molecular recognition(Chen et al. 2011) leading to self-assembled, folded structures made from block-copolymers,(Murnen et al. 2010) biopolymers(Rothmund 2006) or patterned microparticles. Yet none of these techniques have recapitulated the “algorithmic” assembly used by complex organisms to create macroscopic structures.(Peter and Davidson 2009) Very precise submicroscopic structures have been generated using deterministic DNA assembly in so-called DNA origami, but this is at the molecules’ own size scale and is not scalable to cellular length scales (Rothmund 2006). Longer-range ordering has been accomplished with DNA assembled nanoparticle crystals, but the definition of the pattern is limited to repetitive patterns(Macfarlane et al. 2011).

Biological patterns are often an outgrowth of the behavior of reaction-diffusion networks, as first described by Alan Turing(Turing 1952). Mathematical models of reaction-diffusion networks have been shown to be capable of generating complex and beautiful patterns resembling everything from leopards’ spots to variegated pigmentation in sea shells. That said, the first actual demonstration of a biological Turing mechanism occurred almost 40 years after the theoretical description, (Castets et al. 1990) illustrating how difficult these systems are to study, let alone engineer.

One of the aims of synthetic biology is to standardize the engineering of biology. Being able to rationally program spatial-temporal organization would be a great

accomplishment, but requires the ability to algorithmically set down biological molecules and superstructures in specific times and places. While no scalable, programmatic pattern formation has yet been demonstrated, we now describe a practical approach that should allow for arbitrary pattern formation from bottom-up principles. Our approach appropriately rests on having programmable chemical reaction networks (CRNs) unfold in time and space.

While complex chemical reaction-diffusion systems (e.g., the well known B-Z reaction)(Vanag and Epstein 2001) are known, they are far from programmable. We will instead rely upon implementing CRNs with programmable DNA circuits(Yin et al. 2008; Phillips and Cardelli 2009). Soloveichik et al.(Soloveichik et al. 2010) have previously described a method by which CRNs can be implemented in DNA, and some of that system’s predictions have been verified in vitro(Zhang and Winfree 2009). However, this work focused solely on the implementation of DNA CRNs in time, rather than in space. We now hope to design DNA CRNs that are inhomogeneous in space. We will initially focus on small, modular DNA reaction networks that can be treated as “primitives,” meaning that the basic reaction can be duplicated, modified, and run in parallel. These primitives can then be the basis for the design of more complex CRNs in algorithmic pattern generators.

Results

Arbitrary reaction networks can be designed and implemented in DNA

In order to form predictable patterns, we require interacting reaction networks. DNA strand displacement reactions can be used to construct individual reactions with predictable kinetics.¹¹ In the strand displacement reaction, a single-stranded DNA molecule (ssDNA) binds to a hemi-duplex DNA molecule via specific Watson-Crick pairings (toehold). This toehold then initiates strand displacement to form a longer, more stable duplex (dsDNA), with concomitant release of a second single strand (**Figure 1A**). Reversible strand displacement reactions can be similarly designed. Because progression of the reaction is only favorable for complementary DNA strands, parallel reactions occurring concurrently in solution can be designed to be chemically

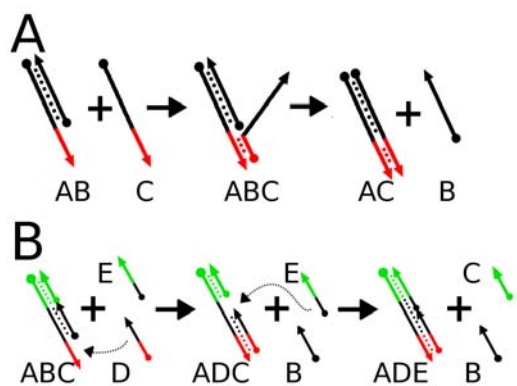


Figure 1: DNA-DNA reactions including. (A) shows a strand displacement and (B) strand displacement chain

“orthogonal,” as eloquently described by Phillips and Cardelli (Phillips and Cardelli 2009).

These strand displacement reactions can be further coupled in arbitrary networks (Soloveichik et al. 2010). Most importantly for our purposes, the individual strand displacement reactions yield single-stranded products that are potential inputs for additional strand displacement reactions. Such coupled reactions can obviously be used to create CRNs. See **Figure 1B** for a schematic of this process. The great advantage of these DNA-based CRNs is that they are rationally programmable, unlike (for instance) a kinase which modifies a transcription factor via a relatively idiosyncratic rule-set in the context of a metabolic CRN. The modularity of the DNA components can be seen both in terms of the flexibility of sequence design, and in terms of the ready combination of components to create the network.

Simulation in MATLAB

The elements of the simulator are diffusion, reactivity with bimolecular kinetics, and a system for displaying the results. These elements can be implemented in either 1D or 2D. Diffusion and chemical kinetics are defined in terms of first order ODE, and can be solved using MATLAB’s ODE45 solver or a simple Euler method solver.

Sequence-mediated diffusion control

In order to create patterns with a reaction-diffusion system,

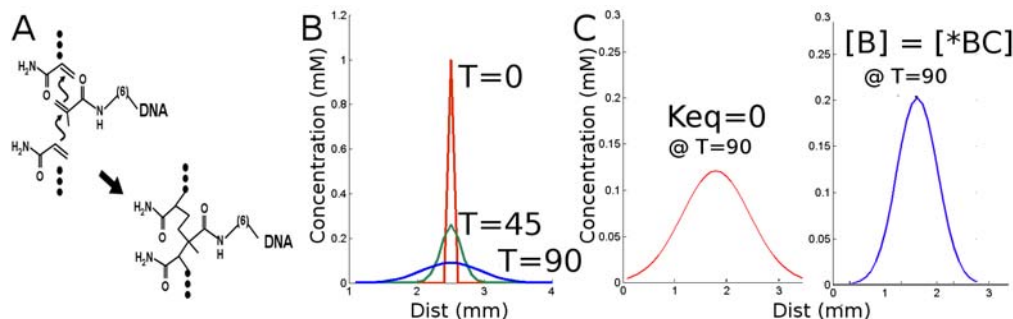


Figure 2: The effect of decorated acrylamide on diffusion. (A) A DNA oligonucleotide terminated with a acrydite moiety is incorporated into a growing acrylamide polymer. (B) Starting from a narrow distribution, DNA spreads through a gel by free diffusion. In (C) free diffusion (left, $K_{eq}=0$) is compared to a species that interacts significantly with the immobile DNA (right, $K_{eq}=0.5$), which diffuses more slowly.

we must control both reactivity and diffusion. We can slow the diffusion of any given component of a CRN system by altering its sequence and affixing antisense oligonucleotides to a hydrogel (for example, by co-polymerizing antisense molecules terminating with an acrylic moiety, an acrydite). **Figure 2A** shows how the DNA may be anchored into the hydrogel superstructure. Depending on the design of a given DNA substrate, gate, or product, some single-stranded DNAs may have partial complementarity to the immobile antisense strand, and others less or no complementarity. This will lead to controllable, differential diffusion through the hydrogel. Diffusion parameters for a given DNA can be altered from fully diffusible to completely fixed depending on the number and strength of the base-pairs formed. This is not unlike chromatography where the equilibrium between bound analyte and unbound analyte determines the retention time.

To work towards the simulation of CRNs in a space where different molecular species will have differential diffusion, we first examine mobile DNA species A and B which are presumed to have equal diffusion coefficients ($D=4 \times 10^{-5} \text{ cm}^2 \text{ sec}^{-1}$) in the gel. In the presence of an immobilized, complementary species *C we compare the predicted diffusion of A and B. We further assume that species A has no significant interactions with *C, but that B does. We can implement this latter reaction as a simple equilibration:



Both C and BC have zero diffusion (noted with asterisk, above) because C is covalently linked to the gel. This slows the effective diffusion of B relative to A (which does not form a complex with *C). Thus the relative diffusion rates of the species differ despite otherwise identical size. To illustrate this, we compare the case where the reaction above is performed under conditions where fast equilibration makes $[B] = [*BC]$ such that species B spends half of its time in a non-diffusive complex. In the case for species A, $K_{eq}=0$ so that species A diffuses freely without interacting significantly with *C. **Figure 2C** shows the results of this simulation. A and B have very different concentration profiles at the same time point. A second modular design element is a short, linear “tail” on the end of other DNA components that partially hybridizes to a stationary, cross-linked molecule to change its diffusion.

Beyond this simple simulation, the diffusion of an

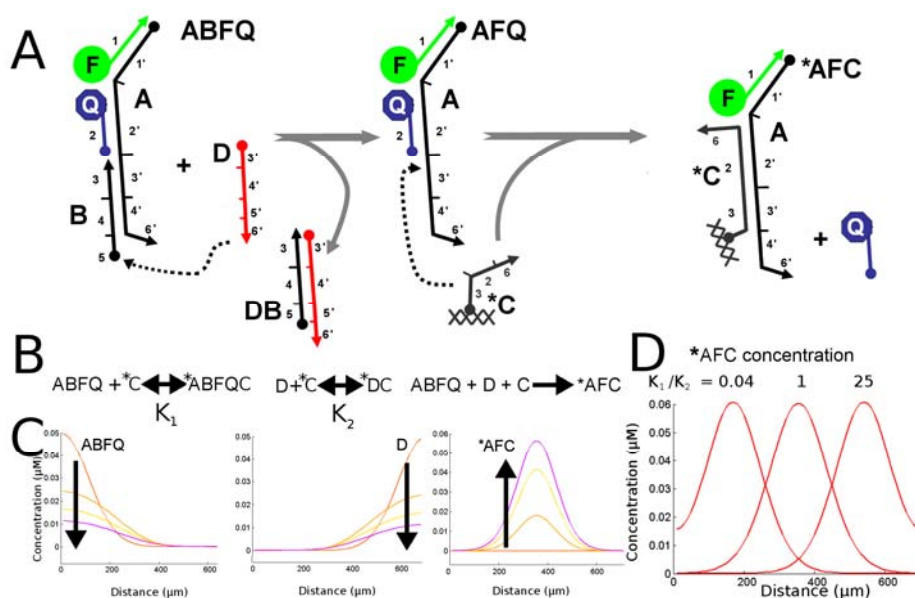


Figure 3: (A) shows the reaction governing the formation of the final, fluorescent product *AFC. DNA sub sequences are numbered with their complementary sub-sequences denoted with an apostrophe. (B) shows the three reactions occurring simultaneously; ABFQ and D are in fast equilibrium with *C to form immobile products (denoted by asterisks) with equilibrium constants K_1 and K_2 . ABFQ and D form an immobile product *AFC when they are co-localized. (C) shows the concentrations of reactants ABF, Q, and D through time as well as the evolution of *AFC. (D) shows the final concentrations of product *AFC as a function of the ratio of the two equilibrium constants K_1 and K_2 .

oligonucleotide is known to be influenced by its size and conformation. This provides opportunities to engineer a given DNA strand's diffusion. We will show that we can implement a spatially controlled reaction by controlling the reactivity and the diffusion of a set of properly designed DNA constructs.

Dynamic modification of diffusion and fluorescence of CRN components

The state of a DNA molecule (i.e. conformation, hybridization) can be transduced to optical information by strategic placement of fluorophores and quenchers, in a manner analogous to a molecular beacon. By having the fluorescence of DNA substrates change as they diffuse, react, and are immobilized we can potentially create dynamic, observable patterns. We will initially illustrate this by having two rapidly diffusing molecules react to form a local, immobilized fluorescent product.

From a historical perspective, this is similar to Ouchterlony double-diffusion experiments (Ouchterlony 1958). In these experiments an antigen and a mixture of antibodies are allowed to diffuse toward each other through a gel matrix. Depending on the diffusion constants of the antigen and antibody, a region of visible immuno-precipitation will occur at some location between the starting locations. Thus, Ouchterlony experiments could be used to infer intermolecular interactions by observing the location of a reaction product.

A similar approach in DNA can be engineered to produce a detectable product at a specific location. The strategy for implementing this is shown in Figure 3A. We set up a simulation modeling two diffusing DNA reagents, ABFQ and D. These reagents interact transiently with an immobilized DNA strand, *C (immobile species are denoted with an

asterisk). This slows their diffusion by a predictable degree as shown above. When they meet at a location between their starting regions, they react and develop a fluorescent product. Fluorogenesis is accomplished by releasing a fluorescent product from its proximity to a quencher. Because the fluorescent product is also complementary to the cross-linked DNA, it is locked in place as it is generated.

Specifying a feature's location by modulating interactions with a DNA-gel

Adjusting interactions with a gel, as we have seen, can change the effective diffusion of a mobile DNA. By tuning the interaction strength, diffusion rates can be specified. These interactions are shown in Figure 3B with their equilibrium constants. By adjusting these equilibrium constants, we can control the location where the fluorescent product is produced. Figure 3C shows the cross section of the fluorescence pattern that would be generated in a gel when both species diffuse at equal rates; product evolution occurs in the center. Figure 3D shows three cases of that result from different ratios of the equilibrium constants, K_1 to K_2 . It should be noted that although the position of product, *AFC, is only affected by the ratio of K_1 to K_2 rather than the absolute values of K_1 and K_2 , these absolute values affect the time required for the pattern to develop.

This clearly shows that the location of the reaction can be varied by changing the relative equilibrium constants which are determined by the degree of complementarity to the immobilized oligonucleotide.

Using complementarity to adjust diffusion

It is computationally expensive to model an equilibrium between fixed and mobile states for each species in the CRN. To simplify and speed up the simulation, we implement an effective diffusion coefficient D_{eff} for each species depending on its assumed complementarity to the fixed DNA. Fast equilibrium with a fixed, immobile state constitutes a time-average of the diffusion at normal rate (corresponding to diffusion coefficient D) and zero. From the concentrations and standard free energy of the DNA-DNA reaction, (Nakano et al. 1999) we can calculate K_{eq} , the standard equilibrium constant for the reaction. And from that, we can derive the fraction of time spent in the fixed state, $*BC$.

First, we calculate the dissociation constant and the concentration of the reactants from their initial concentrations, B_0 and C_0 .

$$K_d = 1/K_{\text{eq}}(1)$$

$$[B] = B_0 - [BC] \quad (2)$$

$$[C] = C_0 - [BC] \quad (3)$$

Taking the definition of the equilibrium constant:

$$K_{\text{eq}} = [BC] / [B][C] \quad (4)$$

And defining the fraction of B bound at any given time as follows:

$$F_{\text{bound}} = [BC] / B_0 \quad (5)$$

We can substitute and simplify using the quadratic equation to express the F_{bound} of BC in terms of B_0 , C_0 and K_d :

$$F_{\text{bound}} = \frac{(B_0 + C_0 + K_d) - \sqrt{(B_0 + C_0 + K_d)^2 - 4B_0C_0}}{2B_0} \quad (6)$$

We can therefore express the effective diffusion coefficient with the following relationship:

$$D_{\text{eff}} = (1 - F_{\text{bound}}) \times D \quad (7)$$

For complex simulations, we will use D_{eff} in lieu of modeling an equilibrium between a mobile and immobile state. To predict the diffusion from sequence, we use an estimated K_{eq} of $B + *C \leftrightarrow *BC$ based on calculated ΔG from widely used base-pair stacking energies (Breslauer et al. 1986).

Specifying location in 2-dimensions using coupled reactions

The ability to specify the location of a reaction product is also expandable into multiple dimensions. The products of two separate, non interaction reactions then proceed to create a third product. In other words, two Ouchterlony line generators can be designed and aligned such that only at the intersection will a final product be evolved. This takes the form of the reactions shown in **Figure 4A-C**. The system shown in **Figure 4A** allows species AC and B to diffuse horizontally and where they meet they produce species C in a vertical line. Likewise **Figure 4B** shows a system that produces a horizontal line of product G. At the intersection of these two lines, a products C and G react sequentially with the immobilized fluorogenic construct $*FPQ$ to form a central

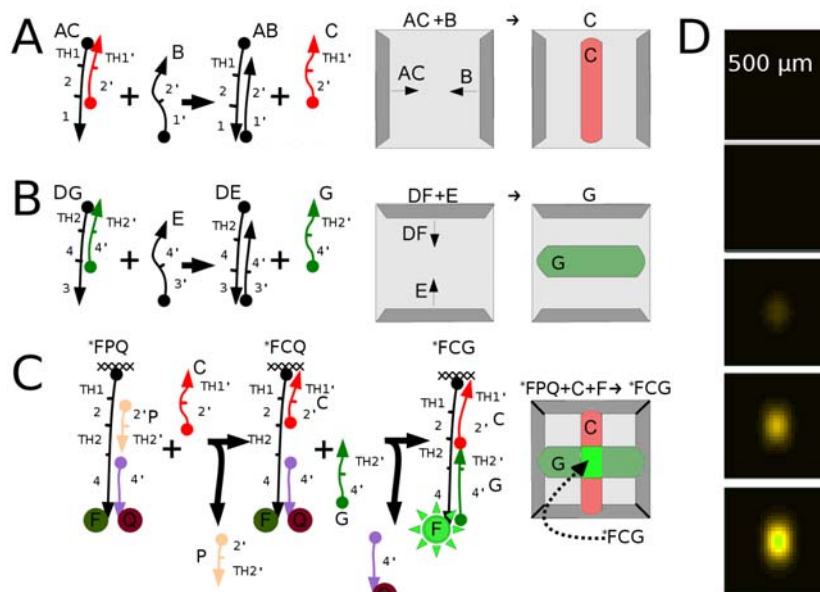


Figure 4: (A) shows a strand displacement that results in product C located in the vertical line. (B) shows a second strand displacement with a different toe-hold that results in product G located in a horizontal line. (C) shows the fluorogenic strand displacement in which immobilized FPQ becomes immobilized fluorescent product FCG only after reacting with *both* C and G. This produces a single fluorescent region located in the center of the gel. (D) shows images from our simulator showing the evolution of product FCG over time; time points are evenly spaced from 10 to 30 hours.

fluorescent spot of the final product, *FCG. In essence, we treated the line generator defined above as a module and applied it again in a second dimension to create a point generator.

Figure 4D shows the results of the simulation of this system. We made the simplifying assumption of using the effective diffusion according to equation 7, alleviating the need for additional terms to account for transiently bound species. The slight asymmetry in the final frame of **Figure 4D** is due to the sequential reaction of X with C then F; the delay introduced by requiring the first reaction to be complete allows for more progress in the reaction originating from the top and bottom reaction and, thus, a taller spot of final product, *FCG.

Reactions can be made orthogonal and used to generate composite patterns

Because DNA-DNA interactions can be designed to be very specific, it is possible to build reaction networks where each reaction is chemically orthogonal; the reactions will not aberrantly interact. Parallel expansion of the 'point generation' program described above allows for the generation of pre-specified, arbitrary, complex patterns via designing the sequence of the interacting DNA molecules. In other words, multiple instances of the type of addressable point generator described above can run within the same gel at the same time and thus form more complicated patterns.

We present an example in which we selectively de-quench immobilized fluorophore in multiple regions where separate, lateral and vertical reaction-diffusion system overlap. Each system, lateral (row) and vertical (column), has a pair of reagents with defined characteristic that determine the final position of the developed feature. A feature therefore can be developed as a "pixel" at an arbitrary position. We call the seven instances of the line generation module A through G. The gel homogeneously includes two immobilized fluorophore-quencher pair species, Xa and Xb. These each require two separate strand displacement reactions in order to become fluorescent. These two threshold regions for the two displacement reactions are shown as TH1 and TH2 in **Figure 4C**.

The process works as follows for reactions systems A, D and G: immobilized Xa has a version of TH1 that responds only to the product of reaction system A. Thus a "primed" column is generated in which TH2 is open only where A reacts. See left column in **Figure 5A**. Products D and G react specifically with TH2 on product Xa (products E, F and H do not). Thus two specific regions of the primed Xa column are fully de-quenched (and turn green). A system of eight such strand displacement reactions per the prototype shown in **Figure 4** can be designed to construct a five point design in 2-dimensions.

With the appropriate diffusion coefficients and reactivities, five regions are induced to fluoresce. The intended regions are shown schematically in **Figure 5A** and the results of the simulation with these parameters is shown in **Figure 5B**.

An interesting property of this system is its scalability. The topology of the pattern will be generated without regard to the dimensions of the gel slab (although the time and material necessary to achieve the result will increase with the size of the gel). This is shown in **Figure 5C** where a simulation was

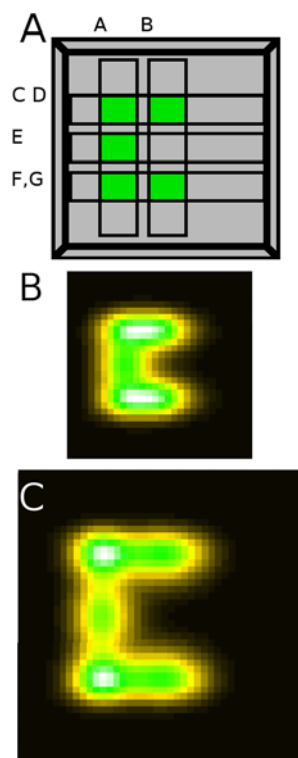


Figure 5: (A) shows the regions activated by the reaction systems described in Table 1. The labeled regions indicate intended location the products of reaction systems A-H should appear. (B) shows a simulation of the same system showing an X pattern. (C) Shows the same simulation with a larger space, all else remaining constant. The pattern scales with the space

run with all parameters consistent to that in **Figure 5B** except that the size of the simulated region was enlarged.

Scalability and resolution limits

The minimal size of the features generated by this system scales with the overall size of the gel in which the reactions are occurring. Minimal features are generated when reactants diffuse only a short distance into each others' territory before reacting. In other words, for sharp features the effective diffusion rate must not greatly exceed the effective reaction rate. Whether a given reaction is diffusion- or reaction-limited can be characterized in terms of the Thiele modulus. Reeves et al. (Reeves et al. 2006) conclude that for effective patterning using diffusing signal molecules, the Thiele modulus must be approximately 1. At this value the influences of reaction and diffusion are balanced.

This can be best illustrated with a thought experiment. We can take a gel of width 600 μm and embed a reaction where two DNA molecules are diffusing towards one another (as shown above in **Figure 3**). They will react with a rate coefficient of $10^6 \text{ mol}^{-1} \text{ sec}^{-1}$ (Soloveichik et al. 2010). If we take the diffusion rate to be extremely slow, the advance edges of the DNA samples will yield a low, broad concentration profile and a correspondingly broad feature (see **Figure 6A**). In the opposite extreme, if we consider a diffusion rate that is very fast, such that the molecules diffuse

across the entire gel before they find a partner and react, then this clearly produces a broad feature (see **Figure 6B**).

In order to find an optimal diffusion constant that will produce a narrow line we can use a numerical simulation. A typical DNA substrate has a native diffusion rate of $4.6 \times 10^{-8} \text{ cm}^2 \text{ sec}^{-1}$ in a 5% acrylamide gel (experimental estimate, data not shown), a value that can be further modified by interactions with oligonucleotides immobilized in the gel. Using the reasonable estimates of the values of diffusion and reaction coefficients above, we estimate that the smallest full-width, half-max (FWHM) feature that can be generated in a 600 μm gel is approximately 63 μm or $\sim 11\%$ of the width of the gel (**Figure 6C**). This corresponds to a hybridization length of 10 residues between substrate and gel, which reduces diffusion by 58% to ca. $2 \times 10^{-8} \text{ cm}^2 \text{ sec}^{-1}$. The topology (and hence the resolution) of the feature size should scale with the outer dimensions of the gel. If we decrease the width of the gel to 150 μm , (see **Figure 6D**) the optimized diffusion coefficient produces a feature of width $\sim 13 \mu\text{m}$ (again $\sim 9\%$ of the width), a modification of the diffusion rate that corresponds to ~ 11 bases of hybridization.

In principle, so long as diffusion can be limited to match the overall size of the gel, there is no limit to the smallest feature size that can be generated except that it will be minimally about 10% of the width of the gel itself. It may be that biological systems that utilize reaction-diffusion systems for spatial organization may be limited in their precision by this same minimal relative feature width. From a practical, experimental standpoint, this implies that to be able to make very small features, one must be able to manipulate increasingly smaller samples. In addition, there will be a

breakdown of the relationship between increased hybridization and lower D_{eff} as a given interaction becomes strong enough to affix DNA strands semi-permanently, so that lateral motion cannot be modeled by simple diffusion. This breakpoint occurs at a k_{off} of ca. 10^{-2} sec^{-1} at room temperature, or approximately 15 base-pairs of interaction between substrate and gel (Robelek et al. 2006). This practical limitation sets the minimal, controllable feature size at about 10 μm for the types of reagents and timescales described here.

From a theoretical standpoint, this work shows that a chemical system can develop an arbitrary feature in space using only chemically defined parameters. There is a resolution limit to systems functioning by passive diffusion. This limit should apply to any CRN that develops spatial order from a homogenous system by such reaction-diffusion mechanisms. Thus, this limitation may be relevant to natural as well as synthetic systems.

Spatial segregation to control chemical reactions

Spatial organization can potentially be used to control chemical reactivity. For example, oligonucleotides linked to small, reactive organic compounds can be organized by templating, which will in turn help to control the order and regiospecificity of the reactions that the small molecules undergo (Li and Liu 2004; Kleiner et al. 2010). By implementing such templated reactions in the context of CRNs it should be possible to add new levels of spatial and temporal control to such assemblies.

We first suggest that control over diffusion can direct the creation of a specific product. Two oligonucleotides carrying reactive chemical species (A and B) can be formed into lines in a gel using procedures those described above (see **Figure 7A**). In this example, the diffusion of precursors AA_0 and BB_0 (see **Figure 7B**) are controlled by partial hybridization of an immobilized oligonucleotide to domains 5 and 6 on the chimeric oligonucleotide precursors. Upon immobilization, strand displacement reactions (e.g. $AA_0 + A_1 \rightarrow A + \text{waste}$) 'activate' A and B to become substrates for additional hybridization and reaction. The diffusion of a third reactant, DNA species D, is similarly adjusted by complementarity to domain 0. Species D diffuses slowly so that it can react with already immobilized, activated A and B, forming either DA or DB. However, since reactive species D diffuses through the activated lines of A and B sequentially this mediates the order of reaction among the small molecule cargoes. Only one of the two possible products shown in **Figure 7C**, DAB, should be generated. We simulate the relative production of DAB relative to DBA in **Figure 7D**.

Similar ordered reactions have been performed by programmed DNA nanorobots (Yurke 2007). However, as with many aspects of the amorphous computations described herein, the scalability of 'classic' DNA nanotechnology is doubtful, especially for the production of chemicals in bulk. Gel-based separations are already common, and thus the concept of controlled, gel-based reactions is more amenable to scaling. Moreover, the process of chemical assembly could occur continuously in the gel, with new reactants constantly diffusing, being activated, and assembling in an ordered fashion. The system is eminently programmable, and changing only the immobilized DNA sequence should change

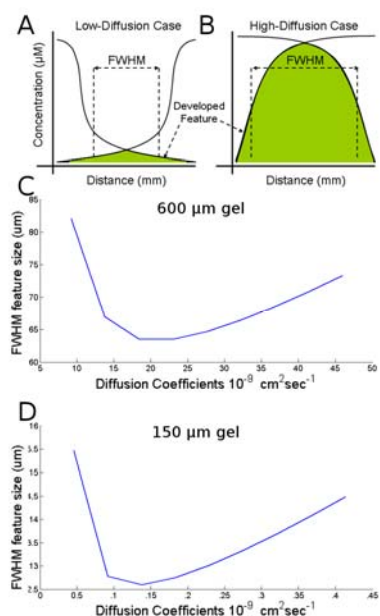


Figure 6: Thought experiment illustration showing that in both the high-diffusion extreme (A) and the low-diffusion extreme (B), the resulting feature is broad. (C) Shows the relationship between the diffusion coefficients of reagents and the final full-width half-max (FWHM) size of the generated feature in the case with a gel of 600 μm in width and (D) 150 μm in width.

the order and kinetics of compound activation, which would in turn alter both the nature and efficiency of production of the final chemical product. Simplistically, depending on the sequences of the chimeras, DBA rather than DAB could be produced with high specificity. More importantly, though, should a given reaction prove inefficient, the width of the self-assembled DNA bands could be increased by simply altering hybridization and diffusivity, allowing more time for the reaction to occur. It is reasonable to think that this system could be combined with programmable nucleic acid reactions¹⁰ to realize a fully programmable, algorithmic system for chemical construction.

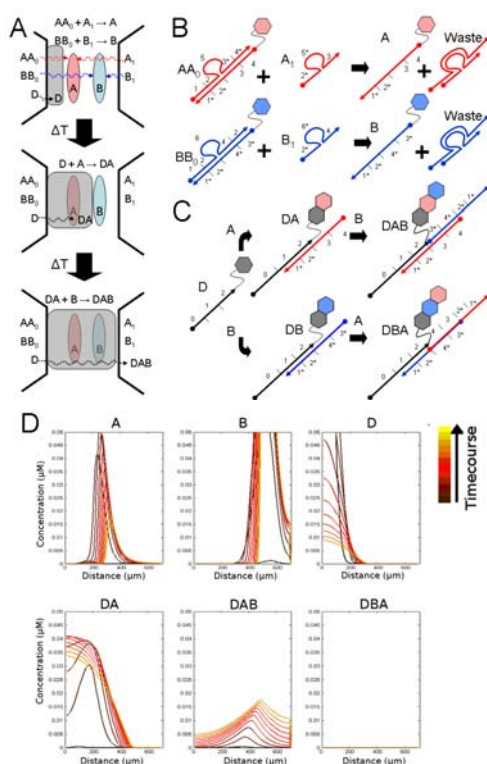


Figure 7: (A) Species A and B develop as lines in a gel through which species D diffuses. (B) Schematic for reactions developing A and B. AA₀ meets A₁ and develops A (and likewise for BB₀ and B). (C) Diffusing DNA species D carries a reactive small molecule (hexagram) slowly through lines of A and B. D and A or B react to form a different product depending on relative locations. (D) Numerical CRN simulation results for all relevant species shows DAB production is greater than DBA.

Discussion

The diverse forms of self-organization in living systems develop from ostensibly simple homogeneity. This has fascinated humans since antiquity (Aristotle 2004). We have suggested a engineerable system that can create spatial patterns from chemical information. Biology excels at this feat but the methods by which it is accomplished are idiosyncratic and not as amenable to engineering as the methods presented here.

Our system is at some level inspired by Alan Turing's seminal paper formulating a set of conditions for pattern formation including plausible kinetic equations with symmetry-breaking properties. Turing speculated that such reaction-diffusion systems could be the basis of embryonic morphogenesis (Turing 1952). His work made clear that specific properties of reactivity and diffusivity are necessary conditions for generating self-organized patterns.

By developing concepts for programming both diffusion and reactivity using nucleic acid sequence information, we provide a path forward for better understanding, mimicking, and ultimately exploiting the CRNs that elude chemists and underlie biology. Biological reaction-diffusion systems are hypothesized to regulate key biological pathways as a general model for the formation of complex patterns (Maini and Othmer 2001). However, despite the long history of research into biological reaction-diffusion systems, most studies focus on either the understanding of natural pattern-formation systems or theoretical possibilities to generate stochastic patterns. Many biological phenomena might be re-imagined in the context of our designable, modular, reaction-diffusion system. Examples might include recapitulating the mechanisms of *Drosophila* development wherein diffusible signals and feedback pathways generate the initial polarization of the embryo. We note that our Turing-inspired simulations predict a resolution limit of 10% of the width of the system. This is indeed the value observed in *Drosophila* development (Gregor et al. 2007). Visibly patterned phenomena such as skin pigmentation may be developed from a reaction-diffusion type Turing mechanism (Nakamasu et al. 2009) and might also be demonstrated with our system. Nucleic acid pattern generators are not yet applicable *in vivo*. Organisms are chemical reaction networks capable of self-replication given appropriate substrates and inputs. Given that there is no adequate definition of life, much less artificial life, our attempts to generate programmable chemical reaction networks can be seen as a first step towards creating synthetic organisms.

Beyond fomenting better understanding of biology, these CRNs should allow entirely new applications in chemistry and materials science. Self-organizing chemistry has previously been experimentally demonstrated in what is now known as the Belousov-Zhabotinsky reaction. This reaction, like Turing's hypothetical reactions, has specific diffusion rates that affect the appearance of patterns (Field and Noyes 1974). However, as was the case with biological development, such reactions cannot be readily elaborated or engineered. New deterministic and algorithmic patterns can potentially lead to the generation of "smart" materials whose bulk architectures are structured down to the nanoscale. For example, Janus particles, whose surfaces are two differentially patterned hemispheres, can be used to generate complex topologies (Chen et al. 2011). It stands to reason that particles with more complex surfaces generated by internal reaction-diffusion systems could generate more complex, patterned associations. Additionally, a reaction-diffusion system might allow for a macroscopic positioning of other DNA structures such as DNA origami (Rothenmund 2006). A meso-scale pattern might be etched into a medium by selectively melting a polymer gel cross-linked by self-assembled DNA helices (Zhu et al. 2010).

In order to rationally design even more complex, algorithmic, developmental programs for later applications, we need to now develop the equivalent of a chemical “compiler” and a test bed for its programs. Complex reaction networks should be specified at a modular level and then rationally rendered into constituent chemicals capable of running the specified reactions. This is only realistic if the design can be made rational and generalized. There must be explicit, computable relationships between sequence and inter-reactivity and between sequence and diffusivity. The thermodynamic properties of nucleic acid hybridization are well known (Nakano et al. 1999). Linear strands with specific energies of hybridization to a immobile strand can thus be computationally designed to specify diffusion. This can be combined with the powerful set of DNA modules that have been shown to be modular has been demonstrably “compiled” into large circuits useful to computation (Qian and Winfree 2011). We envision that the species in such circuits (including amplifiers, thresholds and logic gates) could will dynamically modulate diffusivity by alternately exposing or hiding diffusion-modifying sequences to the fixed medium.

Ultimately, modularity should prove very important in developing such self-organizing systems, as will abstraction and encoding. There is evidence that modularity has emerged from natural evolution as well (Ravasz et al. 2002). By analogy to computer science, implementing a system of modules as an ‘operating system’ for CRNs should be like a high-level computer language. A computer programmer need not know the deepest workings of the hardware (e.g., machine code, register shifts, memory addresses, etc.) in order to write useful software. The work presented herein is a step toward such a CRN language and compiler.

Acknowledgments

This work was funded by the National Institute of Health (1 R01 GM094933 and 1 F32 GM095280), and the National Security Science and Engineering Faculty Fellowship (FA9550-10-1-0169). The content are solely the responsibility of the authors and do not necessarily represent the official views of the sponsors.

References

- Aristotle (2004). *On the Generation of Animals*. Kessinger Publishing.
- Breslauer, K. J., R. Frank, H. Blacker and L. A. Marky (1986). *Proc. Nat. Acad. Sci.* **83**(11): 3746-3750.
- Castets, V., E. Dulos, J. Boissonade and P. De Kepper (1990). *Phys. Rev. Lett.* **64**(24): 2953-2953.
- Chen, Q., J. K. Whitmer, S. Jiang, S. C. Bae, E. Luijten and S. Granick (2011). *Science* **331**(6014): 199-202.
- Field, R. J. and R. M. Noyes (1974). *J. Am. Chem. Soc.* **96**(7): 2001-2006.
- Gregor, T., D. W. Tank, E. F. Wieschaus and W. Bialek (2007). *Cell* **130**(1): 153-164.
- Kleiner, R. E., C. E. Dumelin, G. C. Tiu, K. Sakurai and D. R. Liu (2010). *Journal of the American Chemical Society* **132**(33): 11779-11791.
- Li, X. and D. R. Liu (2004). *Angew Chem Int Ed Engl* **43**(37): 4848-70.
- Macfarlane, R. J., B. Lee, M. R. Jones, N. Harris, G. C. Schatz and C. A. Mirkin (2011). *Science* **334**(6053): 204-208.
- Maini, P. K. and H. G. Othmer (2001). *Mathematical models for biological pattern formation*, Springer.
- Murnen, H. K., A. M. Rosales, J. N. Jaworski, R. A. Segalman and R. N. Zuckermann (2010). *J. Am. Chem. Soc.* **132**(45): 16112-16119.
- Nakamasu, A., G. Takahashi, A. Kanbe and S. Kondo (2009). *Proc. Natl. Acad. Sci. U. S. A.* **106**(21): 8429-34.
- Nakano, S., M. Fujimoto, H. Hara and N. Sugimoto (1999). *Nuc. Acids Res.* **27**(14): 2957-65.
- Ouchterlony, O. (1958). *Prog. Allergy* **5**: 1-78.
- Peter, I. S. and E. H. Davidson (2009). *FEBS Letters* **583**(24): 3948-3958.
- Phillips, A. and L. Cardelli (2009). *J. Roy. Chem. Soc.* **6**(Suppl 4): S419-S436.
- Qian, L. and E. Winfree (2011). *Science* **332**(6034): 1196-1201.
- Ravasz, E., A. L. Somera, D. A. Mongru, Z. N. Oltvai and A. L. Barabasi (2002). *Science* **297**(5586): 1551-5.
- Reeves, G. T., C. B. Muratov, T. Schupbach and S. Y. Shvartsman (2006). *Dev. Cell.* **11**(3): 289-300.
- Robelek, R., F. D. Stefani and W. Knoll (2006). *Phys. Status Solidi (A)* **203**(14): 3468-3475.
- Rothmund, P. W. (2006). *Nature* **440**(7082): 297-302.
- Soloveichik, D., G. Seelig and E. Winfree (2010). *Proc. Nat. Acad. Sci. U.S.A.* **107**(12): 5393-5398.
- Turing, A. M. (1952). *Philos. T. Roy. Soc. B* **237**(641): 37-72.
- Vanag, V. K. and I. R. Epstein (2001). *Phys. Rev.* **87**(22): 228301.
- Yin, P., H. M. T. Choi, C. R. Calvert and N. A. Pierce (2008). *Nature* **451**(7176): 318-322.
- Yurke, B. (2007). *Controlled Nanoscale Motion* **711**: 331-347.
- Zhang, D. Y. and E. Winfree (2009). *J. Am. Chem. Soc.* **131**: 17303-17314.
- Zhu, Z., C. Wu, H. Liu, Y. Zou, X. Zhang, H. Kang, C. J. Yang and W. Tan (2010). *Angew. Chem. Int. Ed.* **49**(6): 1052-1056.

Energy-based Artificial Chemistry Simulator

Vincent Ducharme¹, Richard Egli¹ and Claude Y. Legault²

¹Department of Computer Science

²Department of Chemistry

Université de Sherbrooke, Sherbrooke, Québec, Canada

vincent.ducharme2@usherbrooke.ca

claudel.legault@usherbrooke.ca

Abstract

We present a new artificial chemistry simulator based on simple physical and chemical rules. The simulator relies on a simplification of bonding and internal energy concepts found in chemistry to model simple, large scale, chemical reactions without delay between computation and visualization. Energy introduction and removal can be controlled in the simulations in order to modulate reaction rates. The simulations demonstrate that with this simplified model of artificial chemistry coupled with the concept of energy, it is possible to see the emergence of specific types of compounds, similar to real molecules.

Introduction

The origin of life and the transition from non-living state to living state are fundamental questions that humans have pondered for many centuries. The auto-assembly and self-organization of molecular structures, in particular in biochemistry, are a continuing source of questions and study. The field of artificial chemistry aims to answer these questions. Of course, there are still many longstanding challenges in artificial chemistry. One of them is to demonstrate a model in which the transition to life occurs in Silico (Bedau et al., 2000). The simulator presented in this paper aims at taking a novel route to achieve this goal.

The field of artificial life revolves around simulations to explain phenomena related to the origin of life like emergence, evolution, self-replication and adaptation. Some researchers (Chaumont et al., 2007; Lassabe et al., 2006; Sims, 1994) have developed techniques to create creatures that evolve and cooperate. Using genetic algorithms and neural networks, they created virtual organisms that reproduce intelligent behaviors and execute various tasks. Other researchers are interested in the phenomenon of self-replication (Hutton, 2002; Tominaga, 2005; Hutton, 2007). Using artificial chemistry, these researchers were able to simulate simplified cells that reproduce themselves (Hutton, 2007). They were also able to reproduce biochemical pathways like the fatty acids oxidation (Tominaga et al., 2009). Other researchers went further and made evolving biochem-

ical pathways (Ono and Suzuki, 2003; Hintze and Adami, 2008).

One of the key properties of all living organisms is their ability to reproduce. Researches show that it is possible to obtain simple auto replicative molecules or organisms from artificial chemistry (Tominaga, 2005; Hutton, 2002, 2003, 2005). Dittrich et al. (2001) gives a definition of such artificial chemistry that is a triple $\langle S, R, A \rangle$ where S is the set of particles, R is the set of reactions and A is the algorithm that apply the reactions. Using this definition of an artificial chemistry, it was demonstrated that rules can be specified that allow the replication of some molecules and simple cells (Hutton, 2002, 2007). However, as explained in the third experiment of Hutton (2007), it is necessary to randomly modify the state of the atoms to eventually obtain a cell that could use the defined set of chemical rules to replicate. The property of replication is explicitly defined within the different reactions, thereby limiting the possibility of evolution for the molecules. Emergence is hard to achieve, even if the rules are generic and mutations possible. In order to find the right rules to achieve replication in his chemistry, Hutton created a simulator in the form of a game (Hutton, 2009). The artificial chemistry used in this simulator is based on the same principles used in his previous works. The possible reactions are defined by the user in each level to achieve a specific goal. Even if this method works well to resolve specific problems, the fundamental concept of energy found in physics and chemistry is missing.

However, the simulation of actual chemistry and physics is computationally taxing. Numerous theoretical models exist to describe molecular structures and properties. For example, force fields exist to rapidly describe structural and conformational properties of molecules. These methods can be used on fairly large (i.e. biochemical) systems and molecular dynamics calculations (Van der Spoel et al., 2005). They cannot however describe the electronic properties of the molecules, and reactions cannot be readily modeled. On the other hand, numerous quantum chemistry software packages exist to model electronic properties of molecules with

varying degrees of precision (Schmidt et al., 1993). They are computer intensive, and usually used to obtain single molecule properties. It is clear that formal computational chemistry software cannot be used to rapidly model and study dynamic reactions and emergent phenomena.

Simulating simplified concepts of molecular dynamics and reaction processes of organic chemistry and living organisms is actually more interesting for real-time applications. There are simulators that use this approach. Some of them are based on computer code (Rasmussen et al., 1990; Ray, 1991; Adami and Brown, 1994). Computer code forms programs in the core. When executed, these pieces of code can erase parts of other programs, changing them, thus developing different functional properties. The initial parameters of these simulations allow experiments on different conceptual environment like desert and jungle (more or less resources) and on different type of organisms. Even if these organisms have no equivalent in reality, their behaviors and their properties on the other hand, do. Our goal is to devise a new artificial chemistry simulator that will result in the emergence of dynamic chemical behaviors, through the use of very simple models.

A key concept of our system is to use simple forms of energy to define the chemistry. In Gerlee and Lundh (2010a,b), the energy is used in relation to entropy to determine if an organism can replicate after processing a chain of bits. In our simulator however, a reaction between two atoms will occur if enough energy is involved during the collision. The energy drives and controls the reactions and the evolution of the system. Explicit consideration of energy is in contrast to most rule-based simulators, where reactions occur when two atoms have the right type and a rule to link them.

The next section of this article will give a description of the simulator with the different components and the chemical and physical rules of the system. Results that demonstrate the functionality and viability of the system will be presented in the third section following by a discussion on further development of the simulator.

System description

The system developed can simply and quickly simulate dynamically a large quantity of various atoms. To achieve this, the system is based on an artificial chemistry to simplify calculations. The atoms collide with each other, based on the simple principles of kinematics. They can bind together or break the bonds between themselves by releasing or absorbing energy. The artificial chemistry is simulated onto a two-dimensional grid divided in rows and columns. The borders of the grid can exchange energy with atoms and molecules that collide with it to represent heating and cooling processes. An editor was developed to change the shape of the grid. It allows the specifications of properties for the borders of the container, such as their capability to release or absorb kinetic energy.

Components

The fundamental units of the system are the atoms. They are the components that allow all the interactions and the evolution of the chemistry. As with real atoms, our model atoms possess intrinsic properties: *type* (or element), *mass*, *radius*, and *valences*. The *types* are named after existing atoms $\in \{\text{Hydrogen, Carbon, Nitrogen and Oxygen}\}$; the *mass* and *radius* are also defined to correspond to the actual atomic values. Finally the number of *valences*, which is the number of possible bonds that an atom can do with other atoms, is also defined from known atomic properties. In addition to these immutable properties, the atoms possess *energy*. The energy of the atoms varies throughout the simulation, but remains constant for the entire simulated system in accordance with the principle of conservation of energy. In other words, energy variation on the atoms only occurs through exchange during collisions and chemical reactions.

Group of bonded atoms are called molecules. In the system, a molecule is not a defined entity, it is simply the result of atoms bonded together. Each atom contains its bonding information with other atoms. Molecules do not have any intrinsic properties, besides mass and molecular energy (sum of atomic masses and energies, respectively). These properties are calculated and taken into consideration in the event of collisions. However, since collisions always occur between atoms, either part or not of molecules, the term particle will be used throughout the paper to avoid ambiguities.

The energy of an atom is divided into three categories. *Kinetic energy* represents the energy associated with the motion of an atom in the simulation. It is directly related to its velocity. *Internal energy* represents a crude simplification of the internal vibrational and rotational energies of an atom. In conjunction with the kinetic energy, they are the available energy that an atom can transfer during a collision to break bonds. The last type of energy is the *bond energy*. It represents the electronic potential of an atom. It is the abstraction used to represent the energy stored in electrons to form bonds.

Chemistry and physics

Since we wanted a simulation without delay between calculation and visualization for a large number of atoms, a simplified physics was implemented. Two basic concepts of classical physics are used in the simulation, which are the energy and the momentum conservation.

When simulations start, atoms are positioned randomly onto the grid. To ensure motion in the simulation, they are assigned random velocities. This initialization influences the different collisions scenarios that happen throughout the simulation.

A collision between two atoms will occur only if these atoms are not already bonded together and their centers are at a distance less or equal to the sum of their defined radii, that represents their zones of interaction. Since bonding only

occurs if the two atoms are not already bonded, there are some restrictions on the possible shape a molecule can take. These restrictions will be discussed later.

Energy is a key concept in our simulation. Each atom possesses different kinds of energy. Bond energy represents a simplification of the behavior of electrons involved in chemistry. Bond formation involves pairing of two electrons located on two different atoms. When these two electrons pair to form a bond, they get stabilized, and thus release energy. This energy is transformed into internal energy. Each bond possesses a specific strength that can be defined as *dissociation energy*. It represents the energy required to break that bond and it corresponds exactly to the amount of energy released and transformed into internal energy during bond formation. All the dissociation energies are taken from empirical chemistry tables (Cottrell, 1958). The energy available during a collision thus needs to be sufficient in order to break a bond. The way the atoms collide changes the energy available for a reaction to occur, which is different from a rule-based artificial chemistry.

Using this simple concept, there are four possible scenarios that can occur during inter atomic collisions. Independently of the scenario, there is always a transfer of internal energy between the colliding particles. This transfer is calculated with

$$E_{transfer} = \frac{(E_1 - E_2)}{2} F_{transfer} \quad (1)$$

where E_1 and E_2 are, respectively, the internal energy of the first and second colliding particles and $F_{transfer}$ is a constant transfer factor. This transfer factor represents the ratio of internal energy between the particles. A factor of one means the energy is distributed equally. It is the default value used by the simulator. The internal energy of each particle is then distributed with

$$E'_1 = E_1 - E_{transfer} \quad (2)$$

$$E'_2 = E_2 + E_{transfer} \quad (3)$$

where E'_1 and E'_2 are the new internal energies of each particles following the collision.

An atom with at least one free valence, called a radical, is a highly reactive atom. The first scenario occurs when each of the atoms involved are radicals. When they collide, a bond is automatically formed. There is a release of bond energy representing the bond formation and the electronic stabilization of the atoms. All this released energy is converted into internal energy and distributed to each atoms of the newly formed molecule as a function of their masses and the total mass of that molecule (Fig. 1). Since a bond is formed between the atoms, the resulting velocity of the newly formed molecule must be the velocity of the center of mass of the two particles colliding. This fact is explained by Koenig's theorem, which states that the total kinetic energy

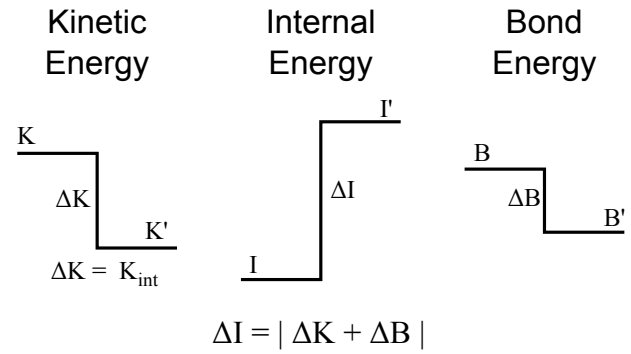


Figure 1: Energies transfers on bond formation. The energies are represented for the whole system in collision. The kinetic and bond energies decrease and are transformed into internal energy.

K_{tot} of a system is the sum of the kinetic energy of the center of mass with the kinetic energy of that system relative to its center of mass, termed *internal kinetic energy* (K_{int}). Since there are only two particles in the collision, the sum can be extended to

$$K_{int} = \frac{1}{2} m_1 \vec{U}_1^2 + \frac{1}{2} m_2 \vec{U}_2^2 \quad (4)$$

$$K_{tot} = \frac{1}{2} m_{tot} \vec{V}_{cm}^2 + K_{int} \quad (5)$$

where m_1 and m_2 are the masses of the first and second particle in collision, \vec{U} are their velocities relative to the center of mass and \vec{V}_{cm} is the velocity of the center of mass. This first scenario is therefore a perfectly inelastic collision. All the *internal kinetic energy* K_{int} is transformed into another form of energy, thus all the resulting kinetic energy is included completely into the velocity of the center of mass. The velocity of the center of mass can be found with

$$\vec{V}_{cm} = \frac{m_1 \vec{V}_1 + m_2 \vec{V}_2}{m_1 + m_2} \quad (6)$$

where m are the masses of each initial particle and \vec{V} their velocities. The *internal kinetic energy* K_{int} is transformed into internal energy. To summarize, when a bond is formed, there is a loss of kinetic energy corresponding to the *internal kinetic energy* of the Koenig's theorem and a release of bond energy representing the stabilization of the atoms. These energies are transformed into internal energy (Fig. 1).

For all three other scenarios, the available energy must be taken into consideration. The available energy is the *internal kinetic energy* (K_{int}) of Koenig's theorem (Eq. 4) for the two particles in collision. To this energy, a part of internal energy of the system in collision is added. This portion is taken in the same proportion as the *internal kinetic energy* (K_{int}) from Koenig's theorem. The energy $E_{reaction}$

required to break a bond is

$$\Delta E = F_{diss} E_{diss} \quad (7)$$

$$E_{reaction} = E_{diss} + \Delta E \quad (8)$$

where E_{diss} represents the *dissociation energy* of the bond and F_{diss} is a constant factor. In the simulator, this factor is set to 10%. This excess energy is used to transfer enough kinetic energy to the two atoms between which a bond is broken allowing them to move away from each other. When more than one bond can be broken, the one with the highest *dissociation energy* is chosen.

An atom without free valence cannot form any bond. When two atoms in that state collide, they can either bounce off each other or break one of their bonds. These are the second and third collision scenarios and are highly related. If the particles do not have enough energy available to break a bond, then the collision is modeled as a perfectly elastic one. The particles just bounce and there is no gain or loss of kinetic energy for the system in collision. Kinetic energy is however distributed normally by the principles of kinematics. The second scenario ends here.

Otherwise, if the colliding particles have enough energy to break a bond, the third scenario occurs. This process can be decomposed in two steps, the first one being a partially inelastic collision. Since the required energy to break a bond can exceed the available internal energy of the system in collision, part of the *internal kinetic energies* of the particles is used to fill the potential well of the bond. As a result, the particles will move away more slowly than before the collision. This is the case a) of Fig. 2. On the other hand, if the required energy to break the bond is less than the amount of internal energy available, the excess of available internal energy is transformed into kinetic energy, resulting in the two particles in collision moving away faster than before the collision (Case b) of Fig. 2). The second step of this scenario is the scission of the bond. Again, the Koenig's theorem is used which also states that the sum of the momentum of each particle relative to the center of mass must be null. Thus, the extra energy ΔE used to break the bond will only be transformed into *internal kinetic energy* for the particles of the scission. The speed of one of the particles relative to the center of mass can then be calculated with

$$\|\vec{U}_1\| = \sqrt{\frac{2m_2\Delta E}{m_1(m_1 + m_2)}} \quad (9)$$

where \vec{U}_1 is the velocity relative to the center of mass and m_1 and m_2 are the masses of the first and second particles. Since the calculation gives only the magnitude of the velocity, an exit angle relative to the center of mass must be set. This angle is arbitrarily set to 30 degrees in the simulation. The final velocity relative to the center of mass can be found with simple trigonometry from the exit angle and the magnitude of the velocity. To find the final velocity \vec{V}_1 of the

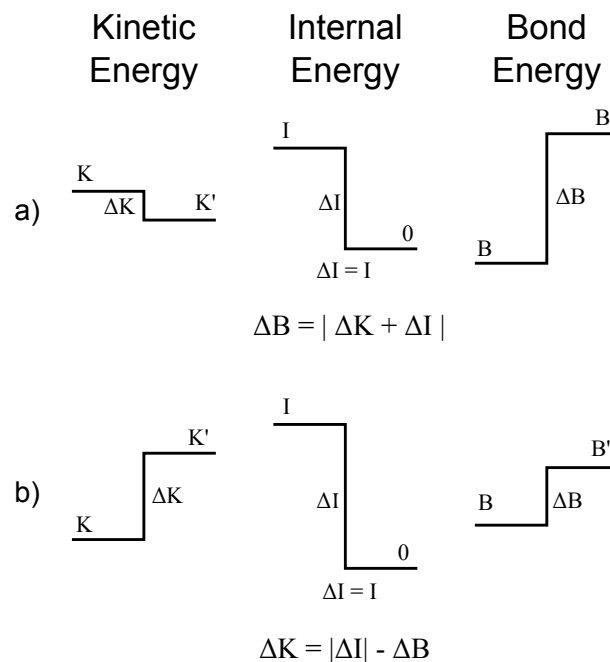


Figure 2: Energies transfers on bond dissociation. The energies are represented for the whole system in collision. There are two possible cases. Case a) occurs when the bond dissociation energy is larger than the internal energy. The remaining energy needed comes from kinetic energy. Case b) occurs when the bond dissociation energy is smaller than the internal energy. The excess is transformed into kinetic energy.

particle back in the simulation frame (instead of the center of mass frame), the result of the elastic collision previously calculated (the bounce between the particles before breaking the bond) that represents the velocity of the center of mass of the molecule that is cleaved must be added to the internal velocity of the particle. Since the momentum must be conserved, \vec{V}_2 can easily be found with

$$\vec{V}_2 = \frac{(m_1 + m_2)\vec{V}_{cm} - m_1\vec{V}_1}{m_2} \quad (10)$$

It is possible for a radical to collide with a stabilized atom. This corresponds to the fourth scenario of collision. When this scenario happens, the simulator breaks a bond from the stabilized atom (scenario 3), freeing a valence. This case can occur only if there is enough energy in the system in collision to break that specific bond (Eq. 8). The newly formed radical is then bonded with the initial radical (scenario 1), stabilizing both atoms again. The atom previously bonded becomes the new radical since it has a free valence. The result is an exchange of atoms between the two molecules. This mechanism keeps the amount of radicals to a reason-

able level. Fig. 1 and Fig. 2 summarize the change in energy for the dissociation and the bonding of two particles.

Molecular Geometry

The way the atoms are positioned around each other will have an influence on their reactivity (i.e. how accessible atoms are for collisions). There are many possible methods to arrange bonded atoms in space. For example, the atoms could simply bond at the position they collide. However, the resulting shape of the molecules in the system could lead to collisions between atoms that should not be possible. Moreover, in reality, atoms bond together in well-defined energy-efficient configurations. To ensure a more uniform representation, all atoms that bond to or break from another are rearranged to represent these energy-efficient configurations. The heaviest atom bonded is used as a reference to decide which one must remain fixed in space and the others are moved according to that position. So, when an atom at the end of a big molecule gains or loses a bond, only the atoms that form the lighter part of the molecule are repositioned. Repositioning the atoms this way facilitates the recognition of molecules from visual information. Two molecules with the same bonded atoms will be represented identically; the only possible difference is the orientation of the molecule in space. Since atoms are not permitted to move relative to each other to simplify calculation, the repositioning is necessary. This also implies that rotational degrees of freedom are not currently implemented.

In the simulator, the carbon and nitrogen have each three valences and oxygen has two. Carbon valences were reduced, instead of the four naturally found, to avoid overlapping problems in the two-dimensional representations. However, to ensure a different reactivity for the carbon atom, the dissociation energies of bonds with the latter are different from the nitrogen atom.

Results

Several simulations were done with the current version of the simulator to evaluate its similarities with respect to classical physics and chemistry, as well as to explore emergent phenomena. The first experiment demonstrates that the system can reach equilibrium in terms of the kinetic energy distribution. The second experiment shows that it can also attain dynamic chemical equilibrium and adapt to a chemical perturbation. The third experiment demonstrates that the simulator enables flexible energy modulations that influence the behavior of the system. Finally, the fourth experiment shows that different molecules can dynamically emerge from the chemistry. The first three experiments use 800 molecules of dihydrogen (H_2), for a total of 1600 atoms. For the last experiment, the number of atoms is set to 800 distributed with 40% of carbon, 20% of nitrogen and 40% of oxygen. Each valence of all atoms is bonded with hydrogen, for a total of 2898 atoms.

First experiment: Statistical distribution of particle speeds

The simulator uses Koenig's theorem to compute kinetic energy distribution between moving particles during a collision. Upon equilibration of a simulation, the distribution of particle speeds should thus obey Maxwell-Boltzmann statistics (for more informations on Maxwell-Boltzmann, see Levine (2008)). For a two-dimensional system, the normalized probability density function is derived as

$$f(v) = \frac{mv}{kT} e^{-mv^2/2kT} \quad (11)$$

where k is the Boltzmann's Constant (defined as 1 in Eq. 11) and T is the virtual temperature of the system. Speed (v) is defined as

$$v = \sqrt{v_x^2 + v_y^2} \quad (12)$$

The average speed $\langle v \rangle$ is

$$\langle v \rangle = \sqrt{\frac{\pi kT}{2m}} \quad (13)$$

The first experiment was designed to confirm this behavior. For the simulations, chemical reactions were deactivated. Simulation was run using 800 H_2 molecules. Molecules were initially given random velocities, but identical speeds. Four simulations were done with initial speeds of 5, 10, 15 and 20. The simulations were each run for 20000 iterations. It takes approximately 1000 iterations to attain stabilization of the speed distribution. At thermal equilibrium (iterations 1000-20000), the speed distribution obeys perfectly Maxwell-Boltzmann statistics, as illustrated in Fig. 3. A theoretical distribution was plotted for an initial velocity of 15. For the simulated distribution, the temperature was defined using the average speed relation

$$T = \frac{2m\langle v \rangle^2}{\pi k} \quad (14)$$

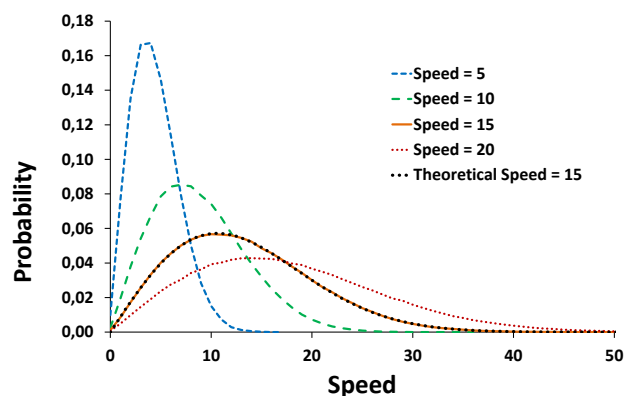


Figure 3: Molecular speed probability distribution. Only one theoretical result is shown to simplify the graph.

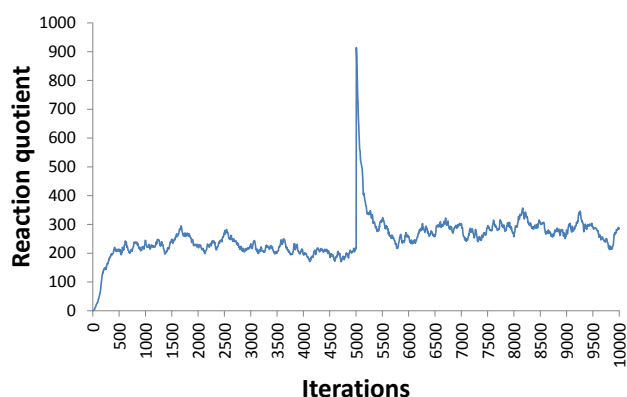


Figure 4: Equilibration of the reaction $2H \rightleftharpoons H_2$. At 5000 iterations, 400 hydrogen atoms were added.

where $\langle v \rangle$ was defined as

$$\langle v \rangle = \sum_i \frac{n_i v_i}{N} \quad (15)$$

where n_i are the relative occurrence of the speed v_i and N the total number of particles.

Second experiment: Dynamic Chemical Equilibrium

The first experiment demonstrates that the simulation, from the point of view of kinetic energies, can attain thermal equilibrium that obeys Maxwell-Boltzmann statistics. The second experiment was thus devised to validate that the simple model for bonding and internal energies, defined in the simulator, could reproduce dynamic chemical equilibrium, as expressed by the Le Chatelier's principle. This principle states that if a chemical system in dynamic equilibrium is disturbed by changing the conditions, the position of the equilibrium moves to counteract and to dissipate the effects of the perturbations (for more informations about Le Chatelier's principle, see Levine (2008)). This simulation was again run with 800 initial H_2 molecules, but with the chemical model activated. The simple dihydrogen dissociation reaction ($H_2 \rightleftharpoons 2H$) was thus studied. The reaction quotient for this reaction is defined as $Q_r = [H]^2/[H_2]$ where $[H]$ and $[H_2]$ are, respectively, the number of hydrogen atoms and the number of dihydrogen molecules. The simulation was run for 5000 iterations. The results (Q_r vs time) are illustrated in Fig. 4. The system reaches chemical equilibrium after 750 iterations, as the reaction quotient becomes stable. To perturb the system, free hydrogen atoms were added after the 5000th iteration and the simulation was resumed. Initially, a drastic increase in Q_r is observed. As the simulation progresses, the system evolves towards a new equilibrium to counteract the perturbation. After 10000 iterations, the system has shifted to a new equilibrium near the

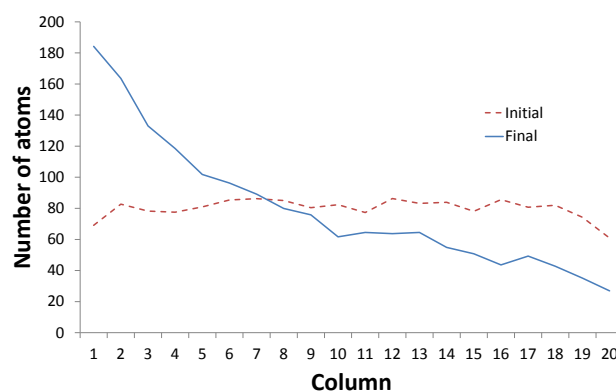


Figure 5: Partitioning of atoms in each column of the simulator grid. The grid has 20 columns and 20 rows. The left column cools the system, the right column heats it.

initial one.

This experiment clearly demonstrates that the system can reach dynamic chemical equilibrium and adapt to perturbations. The simple chemistry model defined in this simulator thus reproduces the Le Chatelier's principle, even if it has not been explicitly defined. It is a result of all the interactions combined with the energy driving the system to a stable equilibrium and it demonstrates, with the results of the first experiment, that the chemistry model is self-consistent and coherent.

Third experiment: Energy influences on the system

As explained previously, the artificial chemistry defined in the simulator is driven by energy (kinetic, internal, bonding) constraints. Modifying the total energy of the simulation should influence the kinetic and chemical behaviors of the atoms and molecules. To show the flexibility of the grid editor and demonstrate that the energy has an influence on the system, the third experiment uses a grid with a side that cools atoms (decreases their kinetic energy) bouncing on it and the opposite side that heats them (increases their kinetic energy). Simulation was run ten times using 800 H_2 molecules with deactivated chemistry. Since the atoms are randomly and uniformly positioned on the grid, the number of atoms by row and column in the grid is initially evenly distributed. After 5000 iterations, a condensation phenomenon is observed on the side that cools atoms. Fig. 5 shows the average initial quantity of atoms per column into the grid and the average quantity after 5000 iterations. The results show clearly that the majority of the atoms are positioned into the left columns.

Fourth experiment: Emergence of molecules

For the final experiment, the simulation involves all atom types defined in the simulator (i.e., H , C , N , and O). The simulation is initiated with predefined proportions of these

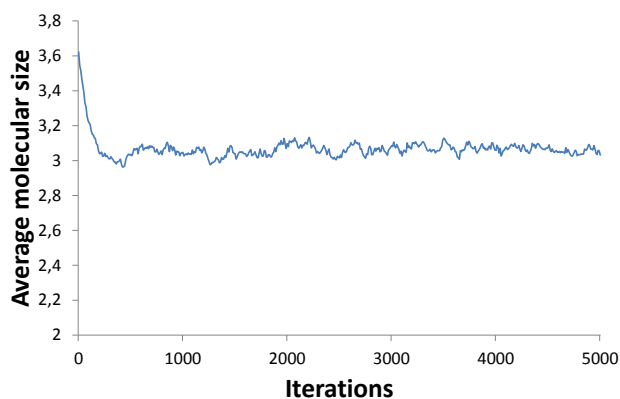


Figure 6: Average molecular size over time. As collisions occur, molecules are broken apart until chemical equilibrium is attained.

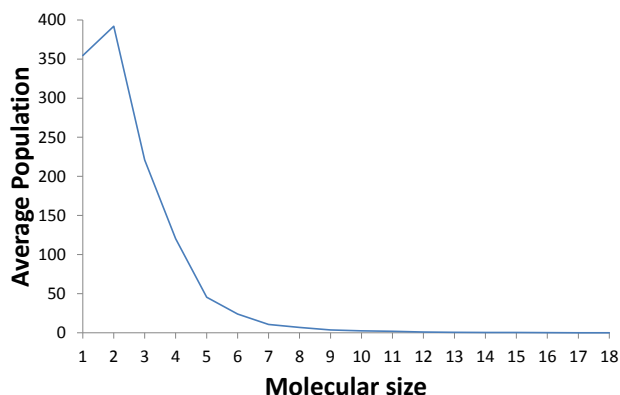


Figure 7: Average molecular population. Smaller molecules tend to accumulate more in the system than bigger ones.

atoms with their valences completely filled with hydrogen atoms (i.e., CH_3 , NH_3 and H_2O). At first, as collisions occur, these initial molecules are broken apart into smaller fragments. From these simple portions, larger, more complex molecules arise. Bigger molecules are inherently more collision prone and therefore, their chance of being broken apart increases, producing more building blocks for others, more stable molecules. With the results of the third experiment, when condensing molecules, the available energy reduces their chances to be broken. After 750 iterations, an apparent chemical equilibrium is attained, and the molecular average size appears to be constant (Fig. 6) which is coherent with the second experiment. Fig. 7 shows the average distribution of molecules with respect to molecular size, regardless of atomic composition. The size of a molecule is represented by its number of atoms. Fig. 8 shows a portion of a simulation. Bigger molecules emerged from simple initial molecules. Dihydrogen molecules have naturally

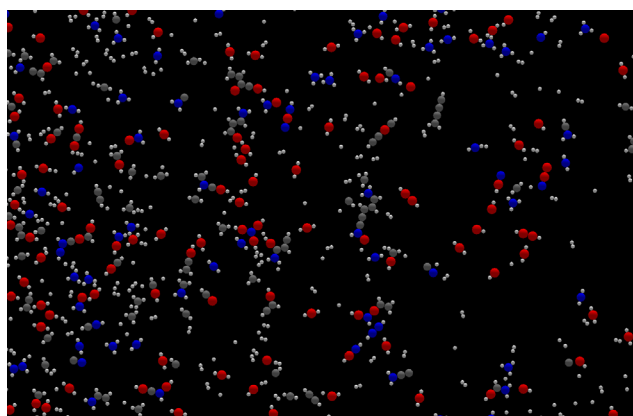


Figure 8: Emergence of complex molecules. Bigger and more complex molecules emerge from simple initial conditions. Hydrogen (white), Carbon (gray), Nitrogen (blue) and Oxygen (red).

appeared from free hydrogen atoms as a result of prior collisions.

Discussion

The simulator and the artificial chemistry described in this work represent a simple, yet reasonable approximation of reality. As demonstrated in the four experiments, realistic physical and chemical behaviors, not explicitly defined, emerged from this simulator. In relation to actual chemistry, properties emerged and were observed. Molecules spontaneously appear from the original blocks and patterns do emerge. The artificial chemistry is therefore constructive.

The transition from simple molecules to self-replicating ones could eventually be achieved with the presented artificial chemistry. At the present moment, there are some restrictions on the type of molecules that can emerge. As explained previously, an atom can only bond to another if they are not already bonded together, directly or indirectly, thus preventing formation of cyclic molecules. This restriction is only due to the current implementation of the control of geometry of the molecules in the simulator, not the chemistry itself. Although cycle formation is interesting, it is not mandatory to observe emergent phenomena; it simply has not been implemented yet in order to speed up the simulator development. It is a feature that will be added in a future version of the simulator.

The simulator currently uses a hard sphere scheme for collisions between atoms. This model is an excellent approximation in the context of the simulator. There are some other schemes that exist to simulate collisions on an atomic scale, like quantum mechanics. Unlike Newtonian mechanics (and thus hard sphere collisions), quantum mechanics is more complex and more computer intensive. Furthermore, there is no need for such precision with the presented arti-

ficial chemistry, because it is only an approximation of the reality, not the reality itself. Another interesting scheme for collision and motion that is valid with Newtonian mechanics is a force-driven simulation. A force-driven simulator could bring interesting add-ons to the artificial chemistry. For example, partial charges could be added on molecules and atoms to influence their motions and positions. With partial charges, surface tension could be modeled as well as hydrophobic phenomenon. This change in the way the atoms move does not, however, influence the specification of the artificial chemistry. Moreover, partial charges could lead to additive non-covalent (non bonding) attractive interactions between molecules, and lead to self-assembly phenomena, bridging the gap between small molecules and biochemical systems.

The actual implementation of motion uses simply the product of velocity and time to move atoms. Since acceleration is null, this method gives excellent results and is accurate. The need for a better integrator will appear only if acceleration changes, for example when forces will be added. However, modification of the simulator will not affect the artificial chemistry and its underlying simple rules.

Conclusion

We have developed a new artificial chemistry simulator that is controlled by the energetic properties of the atoms in the system. With this initial version, it is already possible to observe the emergence of different molecules than the ones involved initially. The use of energy considerations allow a better control of the interactions between atoms than just states and type constraints, and thus represents more accurately actual chemistry. The reaction rules are simple and similar to what is found in nature. We have shown that our simulator is self-consistent, coherent and exhibits emerging behavior similar to chemistry. There are many parameters that can be modified in order to obtain different molecular results and these are what makes the richness of our simulator.

References

- Adami, C. and Brown, C. T. (1994). Evolutionary learning in the 2D Artificial Life system "Avida". In Brooks, R. and Maes, P., editors, *Proceeding of Artificial Life IV*, page 377. MIT Press.
- Bedau, M. A., McCaskill, J. S., Packard, N. H., Rasmussen, S., Adami, C., Green, D. G., Ikegami, T., Kaneko, K., and Ray, T. S. (2000). Open problems in artificial life. *Artificial life*, 6(4):363–76.
- Chaumont, N., Egli, R., and Adami, C. (2007). Evolving virtual creatures and catapults. *Artificial life*, 13(2):139–57.
- Cottrell, T. (1958). *The Strengths of Chemical Bonds*. Butterworths, London, 2nd edition.
- Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial chemistries—a review. *Artificial life*, 7(3):225–75.
- Gerlee, P. and Lundh, T. (2010a). Productivity and diversity in a cross-feeding population of digital organisms. *Evolution*, 64:2716–2730.
- Gerlee, P. and Lundh, T. (2010b). Rock-scissor-paper dynamics in a digital ecology. In *Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*.
- Hintze, A. and Adami, C. (2008). Evolution of complex modular biological networks. *PLoS computational biology*, 4(2):e23.
- Hutton, T. J. (2002). Evolvable self-replicating molecules in an artificial chemistry. *Artificial life*, 8(4):341–56.
- Hutton, T. J. (2003). Information-Replicating Molecules with Programmable Enzymes. *Proceedings of Sixth International Conference on Humans and Computers*.
- Hutton, T. J. (2005). Replicators That Make All Their Own Rules. *Proceedings Workshop on Artificial Chemistry and Its Applications*.
- Hutton, T. J. (2007). Evolvable self-reproducing cells in a two-dimensional artificial chemistry. *Artificial life*, 13(1):11–30.
- Hutton, T. J. (2009). The organic builder: a public experiment in artificial chemistries and self-replication. *Artificial life*, 15(1):21–8.
- Lassabe, N., Luga, H., and Duthen, Y. (2006). Evolving Creatures in Virtual Ecosystems. *ICAT 2006*, pages 11–20.
- Levine, I. N. (2008). *Physical Chemistry*. McGraw-Hill, 6th edition.
- Ono, N. and Suzuki, H. (2003). An Approach to the Evolution of Primitive Metabolic Systems Using a Model of Artificial Chemistry. *ECAL 2003*.
- Rasmussen, S., Knudsen, C., Feldberg, R., and Hindsholm, M. (1990). The CoreWorld : Emergence and Evolution of Co-operative Structures in a Computational Chemistry. 42:111–134.
- Ray, T. S. (1991). An approach to the synthesis of life. In Langdon, C., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Proceeding of Artificial Life II*, page 371. Addison-Wesley.
- Schmidt, M., Baldridge, K., Boatz, J., Elbert, S., Gordon, M., Jensen, J., Koseki, S., Matsunaga, N., Nguyen, K., Su, S., Windus, T., Dupuis, M., and Montgomery, J. (1993). General Atomic and Molecular Electronic Structure System. *Journal of Computer Chemistry*, 14:1347–1363.
- Sims, K. (1994). Evolving virtual creatures. *Proceedings of the 21st annual conference on Computer graphics and interactive techniques - SIGGRAPH '94*, pages 15–22.
- Tominaga, K. (2005). Describing Protein Synthesis and a Cell Cycle of an Imaginary Cell Using a Simple Artificial Chemistry. *Proceedings of the Workshop on Artificial Chemistry*.
- Tominaga, K., Suzuki, Y., Kobayashi, K., Watanabe, T., Koizumi, K., and Kishi, K. (2009). Modeling biochemical pathways using an artificial chemistry. *Artificial life*, 15(1):115–29.
- Van der Spoel, D., Lindahl, E., Hess, B., Groenhof, G., Mark, A. E., and Berendsen, H. J. (2005). GROMACS: Fast, Flexible, and Free. *Journal of Computational Chemistry*, 26:1701–1718.

The Behavior-Based Hypercycle: From Parasitic Reaction to Symbiotic Behavior

Tom Froese¹, Takashi Ikegami¹ and Nathaniel Virgo²

¹Ikegami Laboratory, Department of General Systems Studies, University of Tokyo, Tokyo, Japan

²Max Planck Institute for Biogeochemistry, Jena, Germany

t.froese@gmail.com

Abstract

Most researchers in the science of the origin of life assume that the process of living is nothing but computation in the chemical domain, i.e. information processing of a genetic code. This has had the effect of restricting research to the problem of stability, as epitomized by the concept of the hypercycle and its potential vulnerability against parasites. Stability is typically assumed to be ensured by a rigid compartment, but spatial self-structuring is a viable alternative. We further develop this alternative by proposing that some instability can actually be beneficial under certain conditions. We show that instability can lead to adaptive behavior even in the case of simple prebiotic reaction-diffusion systems. We demonstrate for the first time that a parasitic side-reaction on the metabolic level can lead to self-motility on the behavioral level of the chemical system as a whole. Moreover, self-motility entails advantages on an evolutionary level, thus constituting a symbiotic, behavior-based hypercycle. We relate this novel finding to several issues in the science of the origin of life, and conclude that more attention should be given to the possibility of a movement-first scenario.

Introduction

The scientific debate about the origin of life has traditionally been centered on the competing claims of the ‘replicator-first’ scenario and the ‘metabolism-first’ scenario (e.g. Anet, 2004; Pross, 2004). We have argued extensively that these scenarios are currently in the process of merging into two versions of an information-compartment-metabolism-first scenario (Froese et al., in press). The essential components of the consensus are not new; they are already familiar from Ganti’s (1975) idea of the ‘chemoton’, for example. To be sure, it is commendable that the replicator-first approach is beginning to recognize the value of metabolism, and that the metabolism-first approach is paying more attention to the historical-collective dimension of life. Nevertheless, we have criticized this consensus because it completely ignores the intermediate timescales of life where an individual’s behaviors unfold. Although biologists assume that behavior is a decisive factor for Darwinian fitness in the later stages of life, existing attempts to ground evolution in a prebiotic scenario have tended to focus on chemical factors:

Darwinian competitive exclusion is rooted in the chemical competitive exclusion of metabolism, whether through differential rates of growth or differential resource capture. (Morowitz and Smith, 2007, p. 58)

Indeed, even attempts to generate a more encompassing list of features that could be used to classify the transition from pre-life and life, such as the one compiled by Schuster (2009), fail to even mention the possibility of motility and behavioral interaction. We reproduce Schuster’s list in detail, because it serves as a useful summary of the ideal goalposts of current efforts in artificial life and synthetic biology.

- i. multiplication and inheritance,
 - ii. variation through imperfect reproduction and recombination,
 - iii. metabolism for the production of molecular building blocks,
 - iv. individualization through enclosure in compartments,
 - v. homeostasis and autopoiesis,
 - vi. organized cell division (bacterial cell division or mitosis),
 - vii. sexual reproduction and reductive division (meiosis), and
 - viii. cell differentiation in germ line and soma
- (Schuster, 2009, p. 7)

Schuster’s list is paradigmatic of what we called the new information-metabolism-compartment consensus. Again, it is not that we disagree with the importance of any specific items on this list. But the list as a whole presents an impoverished view of life that neglects the contribution of behavior. We can understand this omission from the standard perspective of the neo-Darwinian synthesis, which integrated biochemistry with population statistics at the expense of ethology. At the same time, however, it should be remembered that even the oldest forms of life, such as the *Archaea* whose lineage dates back to over 3.5 billion years ago, are capable of adaptive behavior including chemotaxis and phototaxis. Indeed, the whole world of single-celled organisms is full of behavior, which suggests that life involved self-motility from the beginning.

Fortunately, a serious appreciation of motility at the origin of life is starting to develop. Although it is widely assumed that intermediate timescales of behavior could not have played a role at the very beginning of life, it has been demonstrated that even simple dissipative structures can exhibit a variety of life-like behaviors (e.g. McGregor and Virgo, 2011; Froese et al., 2011; Virgo, 2011; Hanczyc and Ikegami, 2010; Suzuki and Ikegami, 2009). And there is a small but growing body of research supporting the idea that self-movement and adaptive behavior could have played a crucial role for the origin of life

and proto-cell evolution (e.g. Egbert et al., 2012; Hanczyc, 2011; Froese, et al., in press). In order to distinguish this work from the information-compartment-metabolism framework we refer to it as a ‘movement-first’ scenario.

In this paper we add support to the idea that a movement-first scenario is applicable even in the case of simple prebiotic systems. We address potential criticisms that using a minimal dissipative structure, as an example of a proto-living system, is implausible. For instance, it could be argued that a non-compartmentalized autocatalytic cycle is unsuitable for the origin of life and early evolution, because of (1) a lack of a clearly defined ‘individual’ that could serve as the target of natural selection (Maynard Smith, 1979), (2) a lack of internal functional differentiation for natural selection to choose from (Mossio et al., 2009), and (3) a lack of sufficient protection against the evolution of parasitic side-reactions (Bresch et al., 1980; Maynard Smith, 1979). We have provided an extended response to the first two concerns elsewhere (Froese, et al., in press). Here we focus on the problem of parasitic reactions, because this is one of the most widely discussed issues. The main worry is that prebiotic systems, lacking the selectivity of specialized enzymes, would quickly succumb to side-reactions that receive benefit from the system but do not provide any benefit in return. In the words of Orgel:

The most serious challenge to proponents of metabolic cycle theories—the problems presented by the lack of specificity of most nonenzymatic catalysts—has, in general, not been appreciated. If it has, it has been ignored. Theories of the origin of life based on metabolic cycles cannot be justified by the inadequacy of competing theories: they must stand on their own. (Orgel, 2008, p. 12)

However, this problem may be overstated. Following the pioneering research of Boerlijst and Hogeweg (1991) it has been recognized that spatial embedding and self-structuring of chemical systems plays an essential role in reducing the negative impact of parasites (May, 1991). This has given rise to a tradition of modeling research into what particular aspects of spatiality modify the evolutionary dynamics of populations (e.g. Cronhjort, 1994; Boerlijst and Hogeweg, 1995; Cronhjort and Blomberg, 1997). For example, it was found that spatial self-structuring can constitute a stable composite structure that can serve as a new individual unit of natural selection (e.g. Savill et al., 1997; Hogeweg and Takeuchi, 2003).

Here we push this approach in a novel direction by shifting the current focus from spatial self-structuring to self-generated movement. The upshot of our argument is that the threat of parasitic side-reactions for early proto-metabolic systems may in fact have been overestimated, because the possibility of an adaptive response at the behavioral level of the system has so far been ignored. In brief, *parasites are less of a problem as long as the reaction system tends to move away from them, such as when searching for a more metabolically desirable region of the environment.*

The idea of a chemical system capable of chemotaxis like a bacterium may appear to be implausible, but this behavior has now been demonstrated in different models (e.g. Froese, et al., 2011; Suzuki and Ikegami, 2009) and even in actual chemistry (e.g. Hanczyc and Ikegami, 2010). Thus, while most research is still focused on how spatiality can enhance stability, we are

interested in how instability can be harnessed as a means to do useful behavioral work in space. Using an illustrative example first introduced by Virgo (2011), we show that under some conditions a parasitic reaction on the metabolic level can constitute movement on the behavioral level of the system as a whole, which is adaptive on the evolutionary level.

In the next section we provide some general background to the proposal that current approaches to the origin of life need to be enriched with a movement-first scenario by appealing to a related development in the history of cognitive science. We then take a closer look at one famous proposal for the origin of life, namely the ‘hypercycle’ (Eigen, 1971). On this basis we discuss a simple reaction-diffusion model in order to show that taking the possibility of motility into account is a useful extension to the traditional hypercycle model, thereby leading to the generalized notion of a *behavior-based hypercycle*.

Historical background

Synthetic and molecular biology are largely defined by the assumption that the process of living is essentially nothing but information processing in the chemical domain. Half a century ago a similar idea, namely that the process of cognition is nothing but information processing in the brain, gave birth to cognitive science. What can we learn from its history?

We argue that progress in the science of the origin of life is hampered by a familiar set of misguided assumptions. Just as in the heyday of ‘Good Old-Fashioned Artificial Intelligence’ (GOF AI) and its idealized toy worlds, in today’s molecular biology there is no concern for the requirements and benefits of adaptive behavior in the real world. Indeed, in an implicit agreement with the computational theory of mind, the most widely accepted theories of life are centered on the notion of information processing of symbolic representations, in this case the genetic code. The metabolism-first scenario is only a sub-symbolic alternative to this view, just like sub-symbolic AI was a version of GOF AI that also continued to share the commitments of the computationalist framework.

And just like this computationalist AI had locked the mind inside of the head, synthetic biology (and much artificial life) has constrained life to reside inside a membrane boundary. In recent versions of the RNA-world scenario, for instance, all essential processes involved in the first instances of life are assumed to take place inside of an insulating compartment. This compartment ensures a fundamental division between an internal ‘system’ and an external ‘environment’, where the former is controlled by the genetic system. This insistence on the notion of internal control and on a dualistic distinction between controller and body, as well as between body and environment is, of course, familiar from traditional cognitive science. Even life’s requirement of continuous material and energetic exchange with the environment is conceived of as nothing but a contingent feature of the chemical domain. It is conceptually treated as no different than a robot’s ‘need’ for an external power supply. Accordingly, it is assumed that the process of living can be synthesized and studied in relative disregard of the metabolic body and the environment, which in any case is practically kept as pure and sterile as is possible.

However, as we know from the history of AI and cognitive science, the guiding principles of computationalist AI turned out to be inadequate for the construction of mobile robots that

behaved flexibly and robustly in the real world, especially in environments that were noisy, unpredictable, and fast-paced (Froese and Ziemke, 2009). Although there were examples of successful engineering applications, it became evident that the minds of living creatures must be operating according to other fundamental principles. Eventually the practical shortcomings of classical robotics resulted in a paradigm shift to behavior-based robotics (Brooks, 1991). The assumption that cognition is essentially information processing of abstract symbols was decisively rejected in favor of a treating cognition as primarily an embodied and situated engagement with the world. On this view, mind is a relational phenomenon that emerges out of the distributed dynamics of brain, body and environment (Beer, 1995). The processes of mind are no longer limited to the neural domain of the brain (Clark, 2008). Finally, embodied action became a core concept in the latest developments of an enactive cognitive science (Stewart et al., 2010).

We propose that the science of the origin of life is in need of a similar paradigm shift toward an enactive approach that treats life as a relational phenomenon (Di Paolo, 2009). Life emerges out of the distributed dynamics of a genetic system, metabolism, and the environment. It is primarily a form of goal-directed movement like embodied action.

Synthetic biology can play an essential role in this new endeavor. Previously, robotics made a significant contribution to progress in cognitive science by putting the computational theory of mind to a practical test that turned out to highlight its shortcomings. And given the recent advances in synthetic biology, it is likely that there will be increasing opportunities to practically test out different theories of life as well. In addition, given that synthetic biology still shares some of the core assumptions of the computational theory of mind, it is reasonable to expect that its computational theory of life will also face significant shortcomings as experimental situations become progressively more realistic. Fortunately, we have the benefit of hindsight. We are in a position to learn from the failure of computationalist AI and to draw inspiration from the subsequent development of an embodied and situated robotics and cognitive science. In particular, we emphasize one lesson that may help in understanding the origin of life, namely the role of active movement for embodied and situated agents.

Robots that have been designed according to the principles of GOFAI are easily recognizable by their carefully controlled environment, as well as by their unnatural movements. More importantly, their behavior is inflexible, brittle, and does not degrade gracefully. These undesirable characteristics largely result from an explicit attempt to prevent the messy details of the body and the environment from having any influence on the control system. In contrast, robots that have been designed in a relational manner, in order to properly take advantage of the passive dynamics and material properties of the body and the environment, spontaneously exhibit a surprising amount of robustness and versatility. Moreover, active movement of the sensing body facilitates the self-structuring of sensory flows into perceptual forms (Pfeifer and Scheier, 1999). Movement also increases the resolution of what is perceived, as when the sensation of an isolated tactile contact turns into the complex perception of texture through movement along a surface. Note that the particular structure of a sensorimotor loop is related to the agent's potential for embodied action in a given situation, which grounds its cognition. Accordingly, a mixture of messy

embodiment and situated movement can enhance an agent's behavioral performance, while at the same time significantly reducing the need for a specialized internal control system.

A behavior-based hypercycle model

We will now demonstrate the relevance of this 'movement-first' approach to a specific debate in the science of the origin of life. A fitting starting point are the extensive arguments surrounding the 'hypercycle' theory developed by Eigen and Schuster (e.g. Eigen and Schuster, 1977; Eigen, 1971; Eigen and Schuster, 1978a, 1978b). They identify three requirements for Darwinian evolution by natural selection to take place on the molecular level:

Metabolism. Following the pioneering work of Schrödinger (1944), Eigen and Schuster accept that living systems belong to the general class of far-from-equilibrium systems, and that they maintain that status by means of ongoing degradation and formation of molecular components, i.e. metabolism. It is on this basis that complexity can be generated, maintained, and eventually selected by natural selection.

Self-reproduction. The eligible molecular structures must have the inherent ability of instructing their own synthesis, e.g. they are autocatalytic. Autocatalysis serves to preserve the existing structure of the system, and hence the information it has accumulated. It is on this basis that complexity can be inherited by subsequent generations.

Mutability. Noise ensures that self-reproduction is not 100% reliable, and errors of copying provide the main source of new information in evolution. This ensures that new variants of the molecular structures are made available for selection.

Eigen and Schuster famously showed that the mechanisms of selective accumulation of information involve an upper limit for the number of elements that can be assembled into one genotype, a limit that is inversely proportional to the average copying error rate per element. If this threshold is exceeded there is an 'information crisis': the information that has been accumulated in the evolutionary process so far becomes lost over generations. Accordingly, an increase in the amount of inheritable complexity depends on an increase in the fidelity of genetic transmission.

Eigen and Schuster argue that at the molecular level this enhanced fidelity requires the mutually beneficial *functional linkage* among several autocatalytic or self-reproductive units into one hypercycle. The basic idea is that each autocatalytic component aids in the replication of the next component in a chemical regulatory cycle that is closing on itself. Later on we will modify this basic idea by following the notion of life as an extended process that can incorporate behaviors into its self-constitution (e.g. Di Paolo, 2009; Virgo et al., 2011). We show that a generalized concept of functional linkage enables us to conceive emergent adaptive behavior as another potential form of beneficial linkage, which we denote with the concept of a *behavior-based hypercycle*.

There have been many critiques and elaborations of Eigen and Schuster's original proposal. One important shortcoming was highlighted by Maynard Smith (1979). He pointed out that since each self-reproducing unit within a hypercycle is

assumed to be an independent target of natural selection, they couldn't evolve in such a way that would increase the overall fitness of the hypercycle as a whole on the basis of their mutual cooperation. It is worth quoting Maynard Smith at length, because similar reasoning is still guiding much research into the origin of life today.

How then can a hypercycle evolve characteristics which favour the growth of the cycle as a whole, rather than merely its constituent parts? So long as there is no compartmentalisation, it cannot. For natural selection to act, there must be individuals. (Maynard Smith, 1979, p. 446)

We can now better understand why many researchers insist on the necessity of compartments at the origin of life (e.g. Szathmáry and Demeter, 1987). But as we have argued at length elsewhere (Virgo, et al., 2011), to identify an individual by its external spatial boundaries alone is a misguided. This confuses the organizational limits of the living system as a network of processes with its physical interface. A physically distinct spatial boundary may appear to be important for the structure of an individual, at least from the perspective of the internalist framework of the computational theory of life and mind. On that view, individuation is identical with physical containment. But this is not the case for a relational theory of life and mind, which views individuation as a process and its physical boundary as an interface for exchange. For instance, as we will show in the next section, it is in fact possible for a system of autocatalytic processes to constitute an individuated dissipative structure even without a dedicated compartment, and this individual can be subject to natural selection. Similar results have also been found in related work (e.g. Savill, et al., 1997; Hogeweg and Takeuchi, 2003).

Another influential critique of the hypercycle theory was put forward by Bresch, Niesert and Harnasch (1980). Their model reiterated a worry raised by Maynard Smith. Mutations of self-replicating units that only benefit other self-replicating units in the hypercycle, although beneficial to the hypercycle as a whole, will not be favored by natural selection due to the independent fitness evaluation of the individual autocatalytic units. Instead it is likely that a hypercycle will succumb to so-called 'parasites', i.e. mutant reactions that receive benefit from the hypercycle without providing any benefit in return:

A hypercycle open to new members, i.e. to evolution, is equally open to its killers. How, then, could a hypercycle evolve? Protection could apparently be achieved by spatial separation – be it a wide geographic heterogeneity of RNA populations, a complex formation, or the encaging of entire hypercycles in compartments – a fate, which will sooner or later overtake a hypercycle anyhow. (Bresch, et al., 1980, p. 403)

Out of these three options of protecting a network of self-reproducing units against an invasion of parasites, Bresch, Niesert and Harnasch choose to follow the classical tradition in the study of the origin of life. They claim that a simplified version of a hypercycle must be enclosed as a 'package' in order to evolve in a stable manner. It has been widely debated whether the addition of a compartment can facilitate Eigen and Schuster's hypercycle scenario (Niesert et al., 1981), or if perhaps it can circumvent the need for a hypercycle entirely,

because selection at the level of the compartment is equivalent to group selection of the enclosed, competing self-reproducing units (Szathmáry and Demeter, 1987). In any case, there is a general agreement that a compartment reduces the detrimental impact of parasitic side-reactions (Eigen et al., 1980).

But what about the other two options indicated by Bresch, Niesert and Harnasch, namely a wide heterogeneous spatial distribution and complex formations? These alternatives may not have received sufficient attention, especially considering the difficulty of explaining how several self-reproducing units could fortuitously come to be enclosed inside a compartment so as to give rise to a functioning hypercycle (or some kind of alternative). We speculate that the probability of a successful enfolding would be helped considerably, if there were already a relatively stable network of reactions existing even before the enclosure takes place. In fact, other research has shown that as soon as we move away from models based on ordinary differential equations, and include at least a minimal form of spatial embodiment in an incompletely mixed medium, it is clear that the problem of parasites has been exaggerated in the literature (Boerlijst and Hogeweg, 1991). In some conditions a heterogeneous spatial distribution and/or a complex formation are sufficient conditions for the emergence of group selection and for protection against parasites. Following this tradition, the assumed necessity of pre-biotic compartments at the origin of life must therefore be reevaluated.

In addition, as we will demonstrate, a complex formation can give rise to adaptive behavior at the collective level, i.e. directed movement or 'chemotaxis', which ensures a suitable spatial distribution of the population and thereby reduces that species' vulnerability to local extinction events. In the same model we also demonstrate another possibility that has not yet received sufficient attention. In some cases what looks like parasitic behavior at the metabolic level of the individual self-reproducing units, may instead turn out to be a mechanism of symbiotic behavior when we consider its emergent effects at the level of the system as a whole. The idea that a hypercycle could be conceived of as symbiosis in the chemical domain is not new (Lee et al., 1997). But we extend this idea by showing that this symbiosis can take the form of behavioral interaction in addition to chemical interaction, and that this behavioral symbiosis can even be constituted by parasitic reactions.

The Gray-Scott model

We chose to study a certain kind of dissipative structure that can be found in the Gray-Scott reaction-diffusion system. We use reaction-diffusion patterns because they exhibit some of the essential features of living systems, yet they are easy to simulate and their dynamics can be understood. As with living cells, reaction-diffusion patterns persist by chemically altering their environment, using up available 'food' molecules and temporarily converting them into the substance that makes up their own structure. This process can be thought of as a highly simplified prebiotic metabolism. The 'spot' patterns that can emerge in the Gray-Scott system have the additional property of being composed of many distinct 'individuals', separated by regions in which little chemical activity takes place. We see this process of individuation as analogous to the division of living matter into populations of individual organisms (Virgo, 2011). These individuated spots can exhibit behavior

that depends on the chemical details of their metabolism, as will be shown below.

Although the Gray-Scott reaction-diffusion spots exhibit minimal analogues of metabolism, individuality and behavior, they lack some other properties often associated with living organisms. In particular, they lack specialized genetic material and they lack a physically distinct bounding membrane. This demonstrates that neither a bounding membrane nor the replication of genetic information is a necessary requirement for metabolism and behavior to occur. The existence of distinct individuals despite the lack of a strong separation between their interior and exterior should help to illustrate our point that it might not be necessary for membrane-bound compartmentalization to occur before the onset of evolution by natural selection, even in a metabolism-first scenario.

The Gray-Scott reaction-diffusion system was first studied in a 2D context by Pearson (1993). This is a minimal model of chemical reactions taking place on a flat surface. The reaction modeled is a simple autocatalytic one, $A + 2B \rightarrow 3B$, meaning that when two molecules of B collide with one of A , they react to produce a third molecule of B , while using up one A in the process. A second reaction, $B \rightarrow P$, represents the decay of the autocatalyst into an inert waste product, which is assumed to instantly leave the system. The molecules A and B have a separate concentration at each point on the surface, which are represented by a and b (note that the concentration of P is not modeled). In addition, the ‘food’ molecule A is fed into every point at a rate proportional to $1 - a$. This can be thought of as due to the entire surface being immersed in a solution of A at a constant concentration of 1. In addition to reacting and being added to the system, the chemical species can diffuse across the surface. Overall this gives rise to Equations 1 and 2

$$\partial a / \partial t = D_A \nabla^2 a - ab^2 + r(1 - a) \quad (1)$$

$$\partial b / \partial t = D_B \nabla^2 b + ab^2 - kb \quad (2)$$

where concentrations a and b are functions of space as well as time, r and k are parameters determined by the rates of the two reactions and the ‘feeding’ process (note that the rate of the autocatalytic reaction has been set to 1 without loss of generality). D_A and D_B are the rates at which the molecular species diffuse across the surface. These equations can be solved numerically using a method that is akin to a cellular automaton, except that each ‘cell’ point contains a continually variable amount of the two chemical species.

In this original Gray-Scott model we find different kinds of dissipative structures. Some of these are spatially individuated as self-maintaining spots of autocatalytic chemicals. The spots can divide and replicate. They are also sensitive to gradients of nutrient chemicals, and can react with chemotaxis, although they do not move spontaneously. The spots mutually exclude each other, and therefore during replication will be pushed away from each other. The spots serve as an abstract model of minimal pre-biotic life, but we found them to be limited because they do not have a capacity for open-ended behavior, development, and evolution (Froese, et al., 2011). In a follow-up study we argued that an important but neglected aspect of the pre-biotic scenario of the origin of life was the emergence of motility. We also showed how self-motility could arise in a modified Gray-Scott model (Froese, et al., in press). Here we

continue this research by focusing on what happens to the original Gray-Scott reaction-diffusion spots when they are threatened by the addition of parasitic side-reactions.

We modified the original Gray-Scott model by introducing a second autocatalyst to the system, which feeds not on the ‘food’ molecule but directly on the other autocatalyst (Virgo, 2011). That is, the reactions $B + 2C \rightarrow 3C$ and $C \rightarrow P$ are added to the system, so that Equations 1 and 2 are extended to Equations 3-5, where D_C is the rate of diffusion of C , and k_1 , k_2 and k_3 are the rate constants for the reactions $B \rightarrow P$, $B + 2C \rightarrow 3C$ and $C \rightarrow P$, respectively.

$$\frac{\partial a}{\partial t} = D_A \nabla^2 a - ab^2 + r(1 - a) \quad (3)$$

$$\frac{\partial b}{\partial t} = D_B \nabla^2 b + ab^2 - k_1 b - k_2 bc^2 \quad (4)$$

$$\frac{\partial c}{\partial t} = D_C \nabla^2 c + k_2 bc^2 - k_3 c \quad (5)$$

With an appropriate choice of parameters, the effect of this modification of the Gray-Scott system is to produce the usual spots of the primary autocatalyst, but this time accompanied by a small region of the secondary, parasitic autocatalyst. Since the secondary autocatalyst feeds on the primary one, the spot of primary autocatalyst tends to avoid it by moving away, while the secondary spot follows. This gives the secondary autocatalyst the appearance of being attached as a ‘tail’ behind the primary spot. Thus, the spot-tail system as a whole moves around spontaneously even in a homogeneous environment.

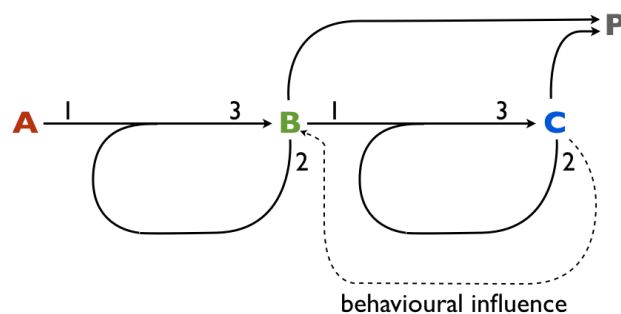


Figure 1. Diagram showing the interactions between chemical species in the reaction-diffusion model. Solid lines represent the chemical reactions $A + 2B \rightarrow 3B$, $B + 2C \rightarrow 3C$, $B \rightarrow P$ and $C \rightarrow P$. The autocatalyst C is parasitic on autocatalyst B . In our simulations, individuated regions of chemicals form, which are composed either out of B or of both B and C . The presence of C changes the behavior of such an individual; it starts to move around spontaneously. This is represented in the diagram by the dotted line marked ‘behavioral influence’.

The spot-tail system is not strictly speaking an autocatalytic hypercycle, because the direct chemical dependency between the two catalysts is not mutual. However, the relationship can still be considered to be an instance of a beneficial functional linkage under some conditions. This is because, although the tail is parasitic on the primary autocatalytic spot (since it does not directly contribute to it metabolically), their co-constituted movement within the environment is adaptive, at least under some conditions. When the chemical interaction (k_2) between the parasite and the spot is strong, a spot can move only in a straight manner and no reproduction occurs. When we weaken

the interaction, the self-moving spots can also reproduce. This transition is important; in the strong interaction regime, self-moving spots will eventually die out by being outcompeted by non-moving spots, but in the weaker regime, the population of self-moving droplets will be sustained by reproduction. With certain parameter settings of the simulation, the spot-tail systems can reproduce more frequently than the spots without tails. Interestingly, at the time of reproduction the self-moving spots can change direction so that they can occupy the whole space. That is, they are more adapted than spots without any ‘parasites’. This is unexpected from the traditional perspective on parasites. It seems that the tail-induced movement tends to split the primary autocatalyst into two distinct spots after some time. Both offspring frequently preserve a tail of their own as well, which means that the trait, once it has been acquired, is passed down the generations like a gene that is transferred from one generation to the next. The movement of the spot-tail systems also tends to make them colonize new areas of nutrients more rapidly. This ‘parasite’-enabled exploratory behavior additionally helps to prevent localized extinction events, in which random areas of the surface are periodically wiped clean of autocatalyst, from eventually killing the whole population (Froese, et al., in press). Here we demonstrated that the spot-tail systems can outcompete spots without ‘parasitic’ tails even in situations that do not include such extinction events (Figure 1).

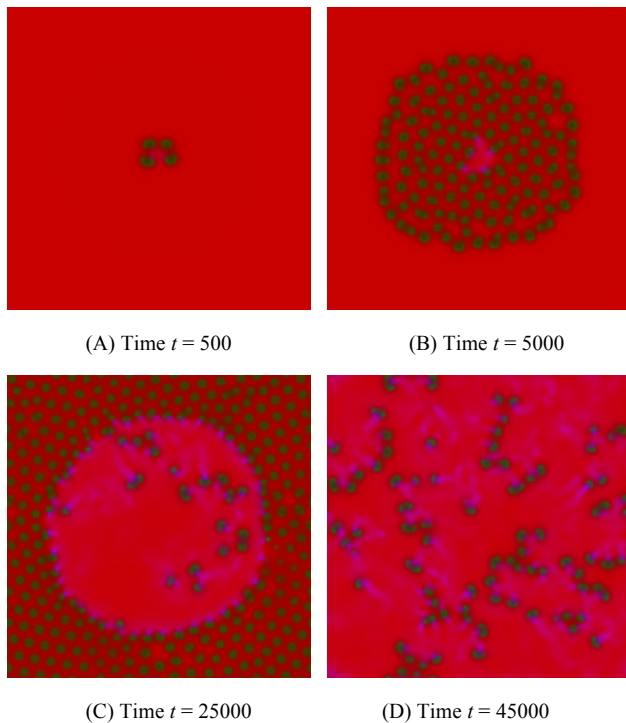


Figure 1. Screenshots of the modified Gray-Scott system with a parasitic side-reaction. At each point the concentrations of A , B , and C are visualized by scaling each of them by 200 and displaying the resulting Red, Green, and Blue (RGB) color value. The surface dimensions are 200 by 200 points. Constants r , k_1 , k_2 , and k_3 are set to 0.025, 0.085, 0.1, and 0.005, respectively. The diffusion rates of A , B , and C are set to 0.1, 0.05, and 0.0025, respectively. Initially, all points are

set to $a = 1$; only in a small 10 by 10 area in the center of the surface are b and c set to small random values drawn from the range $[0, 0.3]$ and $[0, 0.2]$, respectively. (A) At $t = 500$ we can see that the initial seeding of autocatalyst B and parasite C has already given rise to four individual spots that are just about to replicate again. (B) At $t = 5000$ almost the entire surface has been taken over by the spots. In the center we can see that a handful of spot-tail systems have emerged. (C) At $t = 25000$ the remaining outer surface has been occupied by spots. But in the middle there is a growing region in which only spot-tail systems survive. (D) At $t = 45000$ the spot-tail systems have managed to outcompete all of the spots without a parasite.

To be sure, the sequence of events that are shown in Figure 1 is not a necessary result of this modified Gray-Scott system; it is dependent on a certain range of parameters. We have not performed an exhaustive analysis of the parameter space, but we have some practical insights. For example, it is important that the diffusion rate of the parasitic autocatalyst C is significantly slower than the diffusion rate of the original autocatalyst B . Interestingly, in that case the rate of reaction of the parasite C can actually be slightly faster than that of the original autocatalyst B . It seems that the difference in the diffusion processes between B and C is responsible for the break of concentration symmetry, which eventually causes a spot-tail system to move forward. Note that this mechanism of motility is different from the symmetry breaking found in the case of an oil droplet, which is governed by an internal convection flow structure (Hanczyc, 2011). Future work could try to determine more precisely the range of conditions under which relatively stable spot-tail systems emerge.

Previous research about potential benefits of parasites had revealed that their introduction to a model can result in spatial self-structuring (Sardanyés and Solé, 2007). But true benefits have so far remained elusive; to demonstrate symbiosis some researchers relied on the inclusion of catalytic benefit from the parasite to the hypercycle, thereby turning it into a hypercycle by design (Kim and Jeong, 2005). Thus, to our knowledge this model is the first existence proof that a parasitic side-reaction of an autocatalytic system can actually be beneficial in some conditions. This benefit can only be observed when spatiality, self-individuation without containment, and the possibility of movement are taken into account. We can therefore extend the original idea of an autocatalytic hypercycle by including the emergence of system-level behavior as one possible beneficial functional linkage between the chemical components. In other words, the spot-tail system is a behavior-based hypercycle.

Note that this idea of integrating a parasite in order to take advantage of behavioral benefits is not as outlandish as it may appear. For example, the human body can also be seen as a behavior-based hypercycle in just the same way: the brain is metabolically parasitic on the rest of the body, since it uses up metabolites and does not contribute anything back directly on the chemical level. However, it enables us to breathe and find food (i.e. adaptive behavior), and so metabolism of the body is dependent upon the parasitic brain for its own continuation.

Discussion

In order to develop a better understanding of the origin of life we have to pay more attention to all of the various dimensions

and timescales in which this event unfolded. We have insisted on the importance of including more consideration of the role of spatial embodiment and intermediate timescales, because these space-time dimensions are necessary for the emergence of adaptive behavior. More specifically, we contributed to the development of a 'movement-first' approach to the origin of life by evaluating the possible role of movement and adaptive behavior in attenuating the problem of parasitic side-reactions.

A particular challenge for metabolism-first scenarios is the recognition that the first metabolic cycles presumably had to take place without the help of specialized enzymes, which could have significantly enhanced reaction efficiency. Also, it seems that only enzymes could have discriminated between very similar substrates and thus selectively avoided parasites. Accordingly, it appears that a simple dissipative structure, like the reaction-diffusion system in our model, must be especially vulnerable to parasitic side-reactions. It is for these kinds of reasons that Orgel (2008) has argued for the implausibility of metabolic cycles on the prebiotic earth.

It is clear that the existence of a sequence of catalyzed reactions that would constitute an autocatalytic cycle is a necessary condition for the cycle to function in a sustained way, but it is not a sufficient condition. It is also necessary that side reactions that would disrupt the cycle be avoided. [...] Lack of specificity rather than inadequate efficiency may be the predominant barrier to the existence of complex autocatalytic cycles of almost any kind. (Orgel, 2008, p. 8)

Specialized enzymes are clearly an important evolutionary milestone to ensure the increased efficiency and specificity of a metabolic system. However, they are a *necessary* solution only from an internalist perspective on life. On the other hand, if we adopt the relational perspective of the movement-first approach, then an unexplored alternative is made conceivable. We know that if the autocatalytic system is an individuated dissipative structure, such as the reaction-diffusion spots in our model, then the system spontaneously exhibits chemical gradient following, i.e. chemotaxis. The emergence of this self-motility and adaptive behavior is an alternative solution to Orgel's challenge. Chemotaxis (1) enhances the *efficiency* of the chemical reaction by moving the autocatalytic system into regions with higher concentrations of nutrients, and (2) it also enhances the *specificity* of the reaction, because it moves the autocatalytic system away from the negative influence of parasitic side-reactions. In other words, *selective behavior in relation to the environment can partially substitute for the efficiency and selectivity of enzymes within the boundary of a protocell*. Finally, we note that this potential contribution of movement is not restricted to the metabolism-first scenario in as far as the replicator-first scenario is arguably also faced by the same problem of insufficient selectivity due to a lack of specialized enzymes (Shapiro, 2000).

Conclusions

The main points of this paper can be summarized as follows:

- The formation of a pre-biotic individual system does not necessarily require a special compartment; some dissipative

structures are able to self-organize their own spatiotemporal individuation, for instance in the form of chemical gradients.

- These kinds of individuals can exhibit adaptive behavior in an incompletely mixed spatial medium, especially selective self-movement in a chemical gradient (e.g. chemotaxis).
- Chemotaxis reduces the necessity for internal catalytic *efficiency*, such as provided by specialized enzymes, because the individual seeks out regions of its environment that tend to increase its chemical concentrations.
- Chemotaxis thereby enhances an individual's chances of reproduction, because it increases its access to regions that are rich in nutrients.
- Chemotaxis reduces the necessity for internal molecular *selection*, such as provided by specialized enzymes, because the individual avoids regions of its environment that tend to decrease its chemical concentrations.
- Chemotaxis thereby reduces an individual's vulnerability to parasitic side-reactions, because it moves away from any regions that reduce the concentration of its constituents.
- Interaction between an individual and a parasite can give rise to movement of the individual-parasite system as a whole, which in turn is an adaptive behavior in some environments.
- This kind of emergent symbiotic behavior can substitute for a lack of autocatalytic functional closure by constituting a novel behavior-based linking function in a hypercycle.
- A hypercycle that incorporates a behavior-based linking function confers advantages similar to a standard autocatalytic hypercycle; it enhances replicative success of both reactions together and enables group selection.

In sum, the model has demonstrated a novel possibility, i.e. that a parasitic interaction on the metabolic level can result in a symbiotic interaction on the behavioral level of the spatially embedded reaction-parasite system as a whole. It constitutes a new integrated individual that confers evolutionary advantage on the interaction processes of its components. It is easy to imagine that if the evolutionary advantage of such moving-information is strong enough, then the original autocatalytic reaction and the parasitic reaction may eventually evolve to form a proper autocatalytic hypercycle in order to reduce the chances of the parasite killing the host or the host losing its parasite. Chemical endosymbiosis may be an interesting target for future research in this direction.

To be clear, we are not trying to suggest that the Gray-Scott reaction-diffusion system is a realistic model of the origin of life. We have used that system as a proof of concept to show that already extremely simple pre-biotic chemical systems can exhibit individuality, movement, and adaptive behavior – even without a rigid compartment, digital genetic system, or any specialized sensory-motor interface. Arguably, such behavior could have made a significant contribution to resolving some of the problems faced by the earliest forms of life.

References

- Anet, F. A. L. (2004). The place of metabolism in the origin of life. *Current Opinion in Chemical Biology*, 8, 654-659.
- Beer, R. D. (1995). A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence*, 72, 173-215.
- Boerlijst, M. C., and Hogeweg, P. (1991). Spiral wave structure in pre-biotic evolution: Hypercycles stable against parasites. *Physica D: Nonlinear Phenomena*, 48, 17-28.

- Boerlijst, M. C., and Hogeweg, P. (1995). Spatial gradients enhance persistence of hypercycles. *Physica D: Nonlinear Phenomena*, 88, 29-39.
- Bresch, C., Niesert, U., and Harnasch, D. (1980). Hypercycles, Parasites and Packages. *Journal of Theoretical Biology*, 85, 399-405.
- Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, 47(1-3), 139-160.
- Clark, A. (2008). *Supersizing the Mind: Embodiment, Action, and Cognitive Extension*. New York, NY: Oxford University Press.
- Cronhjort, M. B. (1994). Hypercycles versus parasites in the origin of life: Model dependence in spatial hypercycle systems. *Origins of Life and Evolution of the Biosphere*, 25, 227-233.
- Cronhjort, M. B., and Blomberg, C. (1997). Cluster compartmentalization may provide resistance to parasites for catalytic networks. *Physica D: Nonlinear Phenomena*, 101, 289-298.
- Di Paolo, E. A. (2009). Extended life. *Topoi*, 28(1), 9-21.
- Egbert, M. D., Barandiaran, X. E., and Di Paolo, E. A. (2012). Behavioral Metabolism: The Adaptive and Evolutionary Potential of Metabolism-Based Chemotaxis. *Artificial Life*, 18, 1-25.
- Eigen, M. (1971). Selforganization of matter and the evolution of biological macromolecules. *Naturwissenschaften*, 58, 465-523.
- Eigen, M., Gardiner, W. C., and Schuster, P. (1980). Hypercycles and Compartments. Compartments Assists - but do not replace - Hypercyclic Organization of Early Genetic Information. *Journal of Theoretical Biology*, 85, 407-411.
- Eigen, M., and Schuster, P. (1977). The Hypercycle: A Principle of Natural Self-Organization. Part A: Emergence of the Hypercycle. *Naturwissenschaften*, 64(11), 541-565.
- Eigen, M., and Schuster, P. (1978a). The Hypercycle: A Principle of Natural Self-Organization. Part B: The Abstract Hypercycle. *Naturwissenschaften*, 65, 7-41.
- Eigen, M., and Schuster, P. (1978b). The Hypercycle: A Principle of Natural Self-Organization. Part C: The Realistic Hypercycle. *Naturwissenschaften*, 65, 341-369.
- Froese, T., Virgo, N., and Ikegami, T. (2011). Life as a process of open-ended becoming: Analysis of a minimal model. In Lenaerts, T., Giacobini, M., Bersini, H., Bourguine, P., Dorigo, M. and Doursat, R. editors, *Advances in Artificial Life, ECAL 2011: Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems* (pages 250-257). Cambridge, MA: The MIT Press.
- Froese, T., Virgo, N., and Ikegami, T. (in press). Motility at the Origin of Life: Its Characterization and a Model. *Artificial Life*.
- Froese, T., and Ziemke, T. (2009). Enactive Artificial Intelligence: Investigating the systemic organization of life and mind. *Artificial Intelligence*, 173(3-4), 366-500.
- Ganti, T. (1975). Organization of chemical reactions into dividing and metabolizing units: the chemotons. *BioSystems*, 7, 15-21.
- Hanczyc, M. M. (2011). Metabolism and motility in prebiotic structures. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366, 2885-2893.
- Hanczyc, M. M., and Ikegami, T. (2010). Chemical Basis for Minimal Cognition. *Artificial Life*, 16, 233-243.
- Hogeweg, P., and Takeuchi, N. (2003). Multilevel selection in models of prebiotic evolution: Compartments and Spatial Self-Organization. *Origins of Life and Evolution of the Biosphere*, 33, 375-403.
- Kim, P.-J., and Jeong, H. (2005). Spatio-temporal dynamics in the origin of genetic information. *Physica D: Nonlinear Phenomena*, 203, 88-99.
- Lee, D. H., Severin, K., Yokobayashi, Y., and Ghadiri, M. R. (1997). Emergence of symbiosis in peptide self-replication through a hypercyclic network. *Nature*, 390, 591-594.
- May, R. M. (1991). Hypercycles spring to life. *Nature*, 353, 607-608.
- Maynard Smith, J. (1979). Hypercycles and the origin of life. *Nature*, 280, 445-446.
- McGregor, S., and Virgo, N. (2011). Life and Its Close Relatives. In Kampis, G., Karsai, I. and Szathmáry, E. editors, *Advances in Artificial Life: 10th European Conference, ECAL 2009* (pages 230-237). Berlin, Germany: Springer-Verlag.
- Morowitz, H., and Smith, E. (2007). Energy Flow and the Organization of Life. *Complexity*, 13(1), 51-59.
- Mossio, M., Saborido, C., and Moreno, A. (2009). An Organizational Account of Biological Functions. *The British Journal for the Philosophy of Science*, 60(4), 813-841.
- Niesert, U., Harnasch, D., and Bresch, C. (1981). Origin of Life Between Scylla and Charybdis. *Journal of Molecular Evolution*, 17, 348-355.
- Orgel, L. E. (2008). The Implausibility of Metabolic Cycles on the Prebiotic Earth. *PLoS Biology*, 6(1), e18.
- Pearson, J. E. (1993). Complex Patterns in a Simple System. *Science*, 261(5118), 189-192.
- Pfeifer, R., and Scheier, C. (1999). *Understanding Intelligence*. Cambridge, MA: The MIT Press.
- Pross, A. (2004). Causation and the Origin of Life. Metabolism or Replication First? *Origins of Life and Evolution of the Biosphere*, 34, 307-321.
- Sardanyés, J., and Solé, R. V. (2007). Spatio-temporal dynamics in simple asymmetric hypercycles under weak parasitic coupling. *Physica D: Nonlinear Phenomena*, 231, 116-129.
- Savill, N. J., Rohani, P., and Hogeweg, P. (1997). Self-Reinforcing Spatial Patterns Enslave Evolution in a Host-Parasitoid System. *Journal of Theoretical Biology*, 188, 11-20.
- Schrödinger, E. (1944). *What is Life? The Physical Aspect of the Living Cell*. Cambridge, UK: Cambridge University Press.
- Schuster, P. (2009). Origins of Life: Concepts, Data, and Debates. *Complexity*, 15(3), 7-10.
- Shapiro, R. (2000). A replicator was not involved in the origin of life. *IUBMB Life*, 49, 173-176.
- Stewart, J., Gapenne, O., and Di Paolo, E. A. (Eds.). (2010). *Enaction: Toward a New Paradigm for Cognitive Science*. Cambridge, MA: MIT Press.
- Suzuki, K., and Ikegami, T. (2009). Shapes and Self-Movement in Protocell Systems. *Artificial Life*, 15(1), 59-70.
- Szathmáry, E., and Demeter, L. (1987). Group Selection of Early Replicators and the Origin of Life. *Journal of Theoretical Biology*, 128, 463-486.
- Virgo, N. (2011). *Thermodynamics and the Structure of Living Systems*. D.Phil. Dissertation, University of Sussex, Brighton, UK.
- Virgo, N., Egbert, M. D., and Froese, T. (2011). The Role of the Spatial Boundary in Autopoiesis. In Kampis, G., Karsai, I. and Szathmáry, E. editors, *Advances in Artificial Life: Darwin Meets von Neumann. 10th European Conference, ECAL 2009* (pages 234-241). Berlin, Germany: Springer-Verlag.

Checkpoint Orientated Cell Cycle Modeling Issues in Simulation of Synchronized Situation

Jonathan Pascalie^{1,2}, Valérie Lobjois², Hervé Luga¹, Bernard Ducommun^{2,3} and Yves Duthen¹

¹CNRS - IRIT - UMR5505, University of Toulouse, France
{pascalie;luga;duthen}@irit.fr

²CNRS - ITAV - UMS3039, University of Toulouse, France
{valerie.lobjois;bernard.ducommun}@itav-recherche.fr

³CHU de Toulouse, F-31059 Toulouse, France

Abstract

In this paper, the strengths of the checkpoint-orientated modeling in synchronizing cell population are highlighted. Through different experiments, this work shows how to synchronize a population of asynchronous and heterogeneous cells with our proposed model of cell cycle. We will show that the probabilistic modeling undertaken accurately reproduces the dynamics of cell population under specific environmental conditions.

Introduction

Living world daily reveals its complexity. Understanding and assimilating this complexity is of major relevance. With the latest computation capacity explosion, *in-silico* models are positioned to provide new means of studying and exploring complex living systems. Many questions could be tackled with these approaches, specifically when experiments are difficult to address *in-vitro*. System modeling may therefore use fitted methodologies and bottom-up approaches tend to be the general paradigm. They focus on each functional component of the systems and their interactions; and allow to tame the natural complexity and to represent it in model.

Cellular cultures are a set of experiments used by biologists to characterize *in-vitro* specific features of the cell behavior. For instance, in cancer research, the culture is used to evaluate the impact of pharmacological compounds on specific regulatory mechanisms of the cells. Increasing the understanding of the cell cycle is at the heart of cancer research and therefore, the high opportunities foreseen with *in-silico* simulations of cellular systems let think that prospective search of new therapies could be addressed *in-silico*.

In the different fields of computational and molecular biology, the focus on aspects of the cell cycle differs. Molecular biology models focus on the modeling and simulation of the molecular regulatory network of cycline-dependent kinase (CDK) (Novak and Tyson, 2004). These models can be classified into two kinds of models, the discrete model and the continuous ones. Continuous models basically describe

the evolution in the concentration of proteins using a set of ordinary differential equations, whereas discrete models focus on the activation state of each regulatory protein thanks to a predefined genetic regulatory network (GRNs) (Kauffman, 1969; Chavoya and Duthen, 2008). These models have been commonly used to simulate the cell cycle in yeast (Chen et al., 2004; Novak et al., 2001), frog eggs (Novak and Tyson, 1993; Pomerening et al., 2005), fruit flies (Calzone et al., 2007) and different mammalian cells (Aguda and Tang, 1999; Singhania et al., 2011). These models are molecular-based models and do not account for behavioral considerations at a macro-level, their aims being to focus on the regulatory mechanisms.

The other family of models used to simulate cell proliferation is called Individual Cell-Based Models (ICBMs) (Loeffler and Roeder, 2004). These are a subset of the agent-based models. Agent-based models have mainly proved their relevance in the simulation of different complex systems from social networks to the social behavior of hive insects. Basically, individual cell based models come under two classes: cellular automaton (CA) models and off lattice models. On the one hand, CA models are described by a discretization of the proliferative environment in 2-D/3-D evolution grid, and the cell shape is reduced to a lattice site. In this case, cell behavior is composed of the different update rules set up (Moreira and Deutsch, 2002). On the other hand, off-lattice models have the advantages of leaving evolving cells in a continuous media with continuous shapes. They can introduce topological aspects based on *in-vitro* observation or knowledge. This involves high stakes for investigative considerations. The ICBMs have been successfully used to study the pattern formation in multicellular cultures (Galle et al., 2005; Gerlee and Anderson, 2007), avascular tumor growth (Hoehme and Drasdo, 2010) and the spatio-temporal organization of tissues (Drasdo and Loeffler, 2001). These models generally consider the cell cycle as a single time unit decision and the update frequency is the global scheduler of the cell cycle. Basically, this representation does not allow any consideration on the relevance of the major events occurring during progression in the cell cycle phases.

Moreover, ICBMs and hybrid representations with GRNs have been widely used in Artificial Life to study the mechanisms of morphogenesis (Cussat-Blanc et al., 2010; Doursat, 2006). In these studies the cell cycle has to be seen as the cellular behavior with a bio-inspired paradigm.

Whereas molecular-based models accurately express the dynamics of the advancement of cells in each phase of the cell cycle, the individual-based models often do not, due to their meta-description of the cell cycle. Expressing these dynamics reveals interest in simulating *in-vitro* cultures where external compounds are introduced to study their effects in the dynamics of advancement. Our goal is to simulate as closely as possible the population response to an external stress expressing the dynamics of cells progression at a population scale. For that purpose, we use the simplicity of ICBM representations to describe cellular behavior and to introduce temporal considerations thanks to an accurate description of the cell cycle. This approach led us building a hybrid representation of the cell cycle with a hand-coded regulation network and probabilistic-based cellular processes.

In this paper the problem of synchronizing a population of cells is addressed. This consists in activating a specific checkpoint thanks to environmental modifications. Through these experiments, each checkpoint of the model will be specifically activated and the dynamics of the population under these conditions will be analyzed.

Next section introduces the cell cycle model with its biological background. Section 3 shows the different experiments led *in-silico* to synchronize a population of cells. Finally, the last section will discuss the assumptions made for this work and addresses some questions for the next step which is the experimental validation.

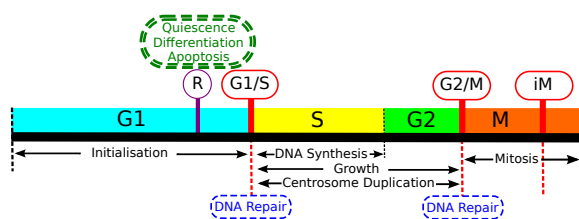


Figure 1: Localization of different cellular processes and checkpoints on the cell cycle timeline. Red simple-lined boxes represent checkpoints with iM being the intra-mitotic one; blue dotted boxes are processes that could be executed during the associated checkpoint; in black with arrows are represented the different processes executed during the cell cycle; the ringed R is the commitment point and the green double-dotted box represents the three exiting points

Cell Cycle Modeling

Biological Background

The cell cycle is often drawn as a circular timeline with different phases starting in G1 and ending at mitosis when a cell divides into two daughter cells. The study of the cell cycle by the biologists puts major emphasis on the essential role of the checkpoints (Elledge, 1996). They are the warrants of the cell's genomic stability and their integrity ensures a good progression on the cell cycle timeline. By the end of the G1-phase, at the commitment point (R), the cell integrates environmental signals before proceeding towards the G1/S transition. A lack of these signals will lead the cell to enter a quiescent (G0) state. If pro-apoptotic signals are detected the cell will undergo death, called apoptosis. Alternatively, differentiation signals will drive the cell out of the cell cycle to a differentiation program. When a cell progresses in the cell cycle, it must accurately duplicate all its internal material (DNA, centrosome etc) and double its mass before preparing for division. Before entering into S-Phase where DNA synthesis occurs, the cell must check for the integrity of its genetic material. This is called the G1/S DNA integrity checkpoint. Providing that DNA synthesis is fully completed, the cell switches to G2-phase and it finishes doubling its mass. During S-phase and G2-phase, centrosome duplication and maturation occurs thus building the two platforms that will allow the assembly of the mitotic spindle required for mitosis to occur. However, before proceeding from G2 to mitosis, the cell must check for the integrity of its genetic material again. This is called the G2/M checkpoint. At mitosis, when cells are dividing, in order to ensure an even segregation of the genetic material into the two daughter cells, the mitotic checkpoint (iM) prevents division until the chromosomes are perfectly aligned on the equatorial plan. Any alteration in these checkpoint mechanisms (for instance a mutation in a key regulator) leads to a genetic instability often associated with transformation and cancer. For these reasons, it is essential to integrate checkpoints as artifacts (or essential milestones) of our simulation model. Figure 1 shows cartography of the cell cycle with the localization of each cellular processes and checkpoints.

In this work the focus of our simulation is put on the temporal behavior of the cells. The checkpoints are the main regulatory mechanism of the cell cycle and are emphasized to study the influence of their activation over a population scale. The modeling process is driven by the temporal problematic. To accurately express the temporal specificities of the cell cycle, the different regulatory mechanisms are described and embedded in a close description of the cell cycle.

The functional and regulatory level of the cell cycle are disjointed. A weakness of traditional approaches in proliferation simulation is often to focus on only one of these aspects whereas the effective cell behavior depends on the interaction between these two levels. Particularly, from

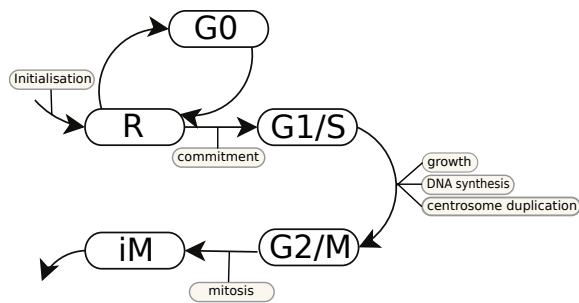


Figure 2: The defined regulators are connected to each other to build a finite state machine (FSM) which embeds the regulatory mechanisms of the cell cycle. The schema also indicates in which position of the FSM are executed each processes.

the cell's internal state depends the regulatory pathway followed. For instance a cell that has replicated its DNA will be allowed to continue its proliferative behavior. To represent these mechanisms and their interaction with the greatest accuracy, it is necessary to observe and describe both levels in accurate cell cycle modeling.

The next part will present how the cellular behavior is designed. This work is based on the model presented in Pascalié et al. (2011).

Cell Cycle Instance

Cellular Processes The cellular behavior introduced leads to the splitting of the cell cycle into sub-behaviors. In this way, each cellular process could be expressed as an autonomous entity. Therefore modeling the checkpoints provides a control over the sequencing of the different cellular processes.

Figure 2 shows the finite state machine designed to schedule the cell behavior. The $R \Rightarrow G1/S \Rightarrow G2/M \Rightarrow iM$ sequence of regulators represents the proliferative behavior of the cells. The cell starts its cycle trying to pass the restriction point (R) and ends with mitosis.

With this modeling approach, the generic cell cycle model designed allows the design of specific cell lineages by instantiating specific checkpoints and processes. The following list describes the different cellular processes. These processes have to be seen as the cell behavior during a transition between two nodes:

- **Initialization:** it represents the G1-phase of the classical cell cycle. During this process the cells have not yet been committed into proliferation, differentiation nor entry into quiescence. This process ends with the transition of the cell at the commitment point. This activity is more a scheduling activity than a functional process of the cell.
- **Commitment:** this action is the planning behavior of the cell. It occurs when the cell has ended its initialization

and when it decides which behavior it will execute.

- **DNA Synthesis:** this activity represents the S-phase of the classical cell cycle. It starts at the end of DNA repair - if necessary - when DNA integrity has been verified at the G1/S transition. During this action the cell replicates its DNA.
- **Growth:** this action represents the cell's mass doubling. It starts at the beginning of the S-phase and ends during the G2-phase.
- **Centrosome Duplication:** this action represents the duplication of the centrosome. It occurs simultaneously with Growth during the S- and G2-phases.
- **Mitosis:** This is the last action of the cell cycle. It requires prior checking of genomic activity at the G2/M transition. If all pre-conditions are met, mitosis occurs in the final stage of the cycle and ends with the beginning of the two new cycles of the daughter cells. Completion of mitosis requires chromosome alignment at the equatorial plan (mitotic checkpoint).

A cell is thus considered to be in G1-phase until it has passed the G1/S checkpoint (if it is executing *initialisation* or *commitment* activities to be precise). A cell is considered in the S-phase while executing *DNA synthesis* regardless of *growth* and *centrosome doubling*. Therefore the cell is considered in the G2-phase when it has ended its DNA synthesis and while it is ending its *growth* and its *centrosome doubling*.

Cell Cycle States The proliferation is not the only behavior observable in this model. The regulatory network presents alternative behavioral functions of the pathway followed by a cell:

- **Differentiation** represents one of the exit points of the cell cycle. If specific conditions are met, the cell will differentiate. This exiting point is available at the R-node (Restriction Point) of the regulatory network.
- **Quiescence**, also named G0-Phase, is an active survey loop used when environmental factors are insufficient for the cell proliferation. The quiescent cells are able to return to the cell cycle at any time if the growing conditions are met. This alternative behavior occurs when the cell is at the G0-node.
- **Apoptosis** represents cellular death. Apoptosis happens if apoptotic factors or signals are delivered to the cell or if the cell spends too much time in a specific stationary situation of its cell cycle. Apoptosis can occur at any time of the cell cycle.

Cell Regulators The pathways previously introduced emerge from a particular sequence of activated checkpoints. The checkpoints are schedulers of the cell cycle in real cells, for that purpose cell regulators are designed and are the warrants of the good sequencing of the processes presented before. These regulators (*i.e* the nodes of the network) are composed of a list of activities along with the preconditions of their activation. They regulate the cell cycle and activate the different processes if their preconditions are fulfilled. If several activities are activated at the same time the cell executes them simultaneously. The preconditions are two sets of boolean flags, one representing the internal state of the cell and the other indicating which activities are done, under progress or planned.

The following list presents the different regulators we defined in our computational cell cycle model:

- The **R commitment point**: cell has to choose between commitment to the proliferation pathway, the quiescent stage, or the differentiation process.
- The **G1/S checkpoint**: here the cell checks its DNA for lesions. If lesions are found, the cell repairs them or die, else it starts DNA Synthesis, Growth and Centrosome cycle.
- The **G2/M checkpoint**: to pass through this checkpoint the cell must have replicated its DNA, should not have detected any DNA damage, have duplicated its centrosome and doubled its mass.
- The **intra-mitotic checkpoint**: to pass this checkpoint and to divide into two daughter cells, the cell needs to have aligned its chromosomes on the mitotic plan and placed its centrosomes on the mitotic spindle poles.
- The **G0 regulator**: we chose to model the G0 state as a regulator because it represents an active survey loop of environmental factors for proliferation. In order to uncorrelate the cell functional level and its regulation, we consider this particular state as a regulatory element of our cell cycle model.

Computational aspects A natural population of cells presents heterogeneous features. Owing to the variability of the duration of each cell cycle phase, two cells born at the same time will not divide simultaneously even if environmental conditions were equivalent. In this work, this heterogeneity is represented with a specific set of parameters for each cell. Therefore, the embedded parameters are generated according to a distribution law. The cell cycle model is thus able to produce a population of specific cells and not only a population of clones. If the cell population was composed of clones, the system would suffer from phasing and synchrony in the sequencing of the different phases, each sister cells going to division at the same time.

To represent the cellular activity in a temporal manner and remain at a macroscopic level of representation, we based the cellular process modeling on their scheduling. In this context, 3 parameters are used for each cellular process: the optimal time of realization, the maximum time before it eventually results in the cell's death, and the probability of success. Using these parameters, we generate a set of parameters which are used for the computation. Our processes are represented over time as Bernoulli processes. The average optimal time determines the number of successes needed to consider the process as achieved and the success rate is used to define the probability of success of one trial.

The simulations processed with this model are discrete-time simulations. The simulation time step is defined at the setup and is fixed to six minutes in the different experiments presented here. At each time step, the agents are randomly sorted and their behavior is processed. At each time step, a bernoulli experience with the parameters previously introduced is intended by each cell and the success of the different experiences lead the cell through its cycle. If a division occurs the divided cell is removed from the population and is replaced by two daughters cells. The parameters of the daughters cells are different. This ensures that the population will not converge to a population of clones. Nevertheless, the daughter cells inherit the DNA lesion of their mother if division could have occurred with it.

The multi-agent system built with the previous elements will be used to validate our cell cycle model with experimental data. The next part is dedicated to the simulation of cells population in synchronized situation.

Experiments

In Pascalie et al. (2012), the qualitative aspects of the cell cycle model were presented. The results shown highlight the ability of the simulator to reproduce specific features of the cell proliferation. The simulation of the exponential growth phase was achieved using specific environmental features and the results presented here use the same specific conditions.

The aim of the work reported in this paper is to demonstrate the model ability to accurately reproduce an important feature of the regulation of cell proliferation that is the activation of specific cell cycle checkpoints. To reach that goal, four virtual synchronization experiments have been performed, each of them leading to the activation of a specific checkpoint. In *in-vitro* experiments, cell cycle synchronization is used to analyze the progress of a cell population through the different stages of their cycle. In this work, the first experiment aims at activating the restriction point (R) avoiding the cell commitment in the cell division cycle by suppressing growth factors from the the environment. The second experiment aims at activating the DNA-damage dependent G1/S and G2/M checkpoints. To achieve this, the deleterious consequences on the genetic material of ionizing

radiation exposure is simulated by defining DNA damage. Similarly, the third experiment aims at selectively activating the DNA-damage dependent G2/M checkpoint. The last experiment will evaluate the intra-mitotic checkpoint activation by simulating an alteration of the mitotic spindle assembly through a well-known procedure known as nocodazole block. Nocodazole is used to disrupt the reorganization of the microtubule network that is required to form a mitotic spindle and therefore leads to the activation of the mitotic checkpoint. This results in cell cycle arrest, thereby synchronizing the cell population at mitosis. All the results presented in this section are the average of 8 instances of simulation. This choice was made to minimize the artifact induced by the pseudo-random number generator used.

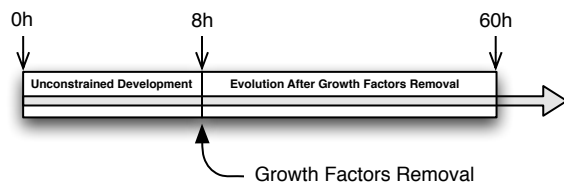


Figure 3: Timeline of the scenario dedicated to analyze the R checkpoint response to growth factors removal. The other experiments introduced in this paper use an equivalent timeline with a different compound introduced at $t = 8h$.

The first analyzed checkpoint is the restriction checkpoint (R). At this point, the cells have to decide whether they are prone or not to commit in the cell division cycle. This decision depends on the availability of various required environmental cues such as growth factors. With this aim a growth factors removal scenario was designed. Figure 3 shows the timeline of this scenario. After 8 hours of unconstrained development, the cells are submitted to a growth factor removal. This change in growth media condition is simulated thanks to a simulation event where the availability in growth factors is set to 0%. Figure 4 shows the result of the simulation. In (a) the evolution of population in each phase shows that after 8 hours of treatment, the cells start to accumulate in G1-phase. The RT_{max} parameter defines the time a cell can spend at the commitment point before entering into quiescence. This parameter is revealed by the results, it consists in the time elapsed between the growth factor removal and the cells entry into quiescence. The population dynamics shown in this experiment is consistent with the common results in this case.

We next analyzed the DNA-damage dependent checkpoints that are activated at G1/S and G2/M transition, when the integrity of the genetic material has been impaired, for instance after exposure to ionizing radiation. These checkpoints warrant the genomic stability in the proliferation of the cells. At these points, the cells have to test their DNA integrity before starting to duplicate it or to proceed into mi-

tosis. To test this response, the cells are virtually exposed to ionizing radiations that lead to DNA lesions. This parameter is integrated in the model and DNA lesions are represented as a DNA injury rate. In this case, the value is set to 100% and the repairing ability of the cells is avoided. As all the parameters of the model, it represents an average value at the population scale. As in the previous experiments, cells are let proliferate for 8 hours in exponential growth phase. Once this time elapsed, the event simulating the UV exposition occurs and the cells receive DNA lesions. Figure 5 shows the result. As in the previous experiments, cells are in exponential growth phase during the first hours. On curves (a), it is noticeable that once the event representing the ionizing radiation occurs, the ratio of cells in S-Phase and Mitosis starts to decrease whereas it increases in the G1-Phase and in the G2-Phase. This is consistent because the cells can neither exit the G1-Phase nor the G2-Phase due to their DNA injuries on the one hand, and the cells exiting the S-Phase and the Mitosis respectively enter into G2-Phase and G1-Phase on the other hand. The decrease of the ratio of cells in G1-Phase and G2-Phase occurs when the cells start dying due to too much time spent trying to pass the activated checkpoint. Figure 5 curve (b) represents the evolution of the population size. The exponential growth phase is characterized by the increase of the population size at the beginning of the simulation. The population size starts decreasing when the cells start dying due to their DNA damages.

In order to refine this analysis, we next simulated the cells' response to the single activation of the G2/M DNA-damage dependent checkpoint. This is a classical situation that occurs in cancer cells that have lost, through the mutation of an essential suppressor gene called p53, the ability to arrest at G1/S upon DNA injury. To perform this simulation we virtually exposed a p53 deficient population of cells to ionizing radiation and examined the consequences of this exposure. Actually, the model does not allow to represent directly this kind of cell lineage. To this purpose, the DNA integrity test that occurs at the G1/S transition is deactivated in an ad-hoc manner for this experiment. This problem will be addressed in the further works section.

Figure 6 shows the results. On these curves the G2 accumulation is observable and it fits with the expected behavior. On the second curve (b), representing the evolution of the population size, the results are consistent. The population stops increasing once entry into mitosis is inhibited by the activation of the G2/M checkpoint and starts decreasing once the cells have spent too much time at this stage and start dying.

The last checkpoint we attempted to simulate was the intra-mitotic one. Nocodazole is used to disrupt the reorganization of the microtubule network that is required to form a mitotic spindle and therefore leads to the activation of the mitotic checkpoint. This results in cell cycle arrest, thereby synchronizing the cell population at mitosis. To simulate the

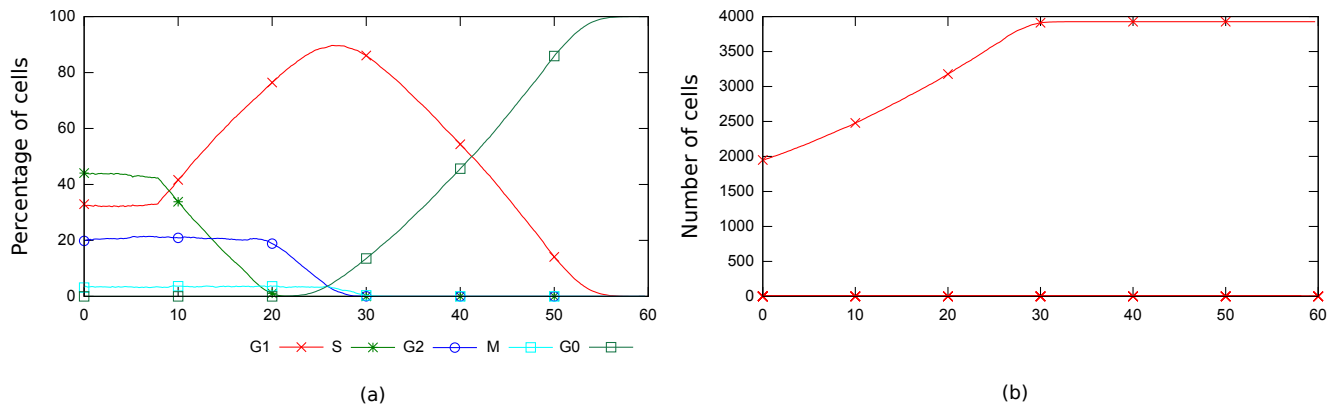


Figure 4: Results of the *in-vitro* experiment aiming the R checkpoint activation. (a) Evolution of the ratio of cells in each phase. Once the growth factors removed the cells start to accumulate in G1-Phase (b) Evolution of the population size. The population still increases after the growth factors removal until all the cells are arrested at the commitment point. The population size does not decrease because the cells enters into quiescence in this case.

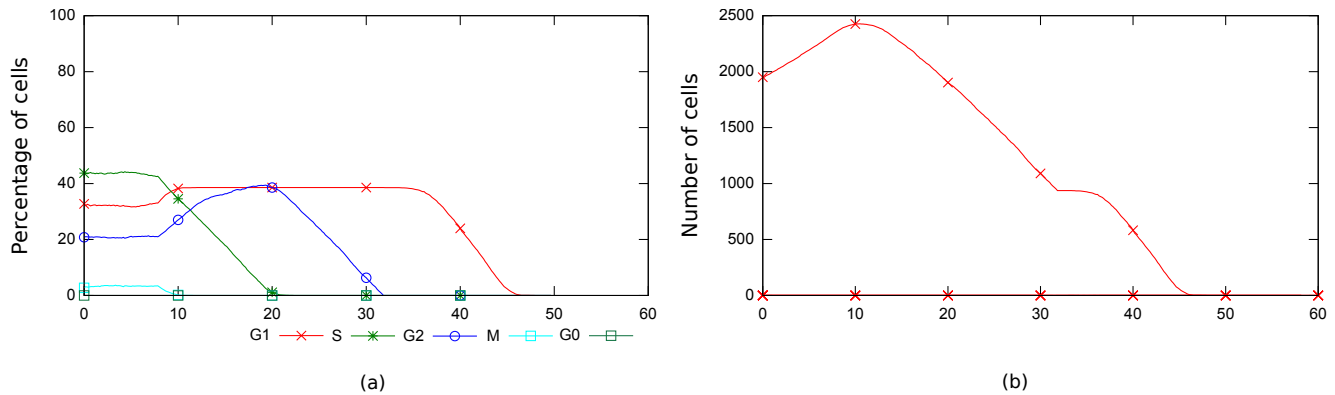


Figure 5: Results of the *in-silico* experiment aiming the simultaneous activation of G1/S and G2/M checkpoint. (a) Evolution of the ratio of cells in each phase. At $t = 8h$ ionizing radiation are induced and the cell proliferation stops. The cells are arrested in G1-Phase while they accumulate in G2-Phase by exiting the S-Phase. They start to die when they have spent too much time trying to repair their DNA damages. (b) Evolution of the population size. The population size stops to increase once the cells are exposed to ionizing radiation .

nocodazole adjunction, we set the mitosis rate of success to $\tau_M = 0\%$. With this parameter the cell will enter in mitosis but should not complete it due to a too small success rate. Figure 7 shows the results of this simulation. On the first stage of the simulation the evolution of the cells in each phase remains constant while the cells proliferate in exponential growth phase. When the event occurs, it is observable that the evolution of the cells in Mitosis starts increasing. This evolution is consistent with the nocodazole effect, which is to affect the mitosis machinery and thus avoid the mitotic spindle formation.

Discussion and Further Works

The interest of the work presented here resides in the ability of the *in-silico* model to accurately reproduce the cells' response to environmental modification and to the activation

of cell cycle checkpoints. The probabilistic modeling undertaken here allows to accurately reproduce the qualitative aspects of the cells dynamics. Nevertheless this approach undergoes a lack, which is the difficulty in parameters tuning due to the difficulty to map experimental biological values to probability. In some cases, it is conceivable that the search for the best values needs to carefully analyze the model response under different conditions in order to map biological observation to a particular setup. It is conceivable to automatize this search using evolutionary strategies. With a set of relevant *in-vitro* data, the best value, and moreover the best scenario fitting with these data, could be found.

The use of this model could be extended to research new compounds or to determine which cells interactions have to be highlighted to answer a particular need. It is conceivable that a scenario should be determined thanks to genetic

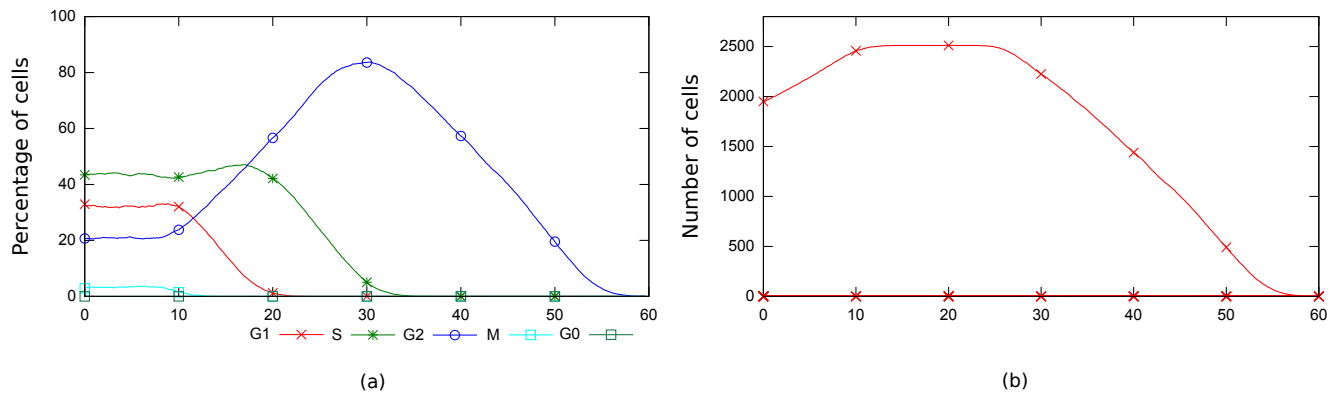


Figure 6: Results of the *in-silico* simulation aiming to solely activate the G2/M checkpoint. (a) Evolution of the population in each phase. The cells start to accumulate in G2-Phase once the ionizing radiation occurs. The G1-Phase accumulation is not observed because the G1/S DNA-integrity test has been deactivated. (b) Evolution of the population size. Once all the cells have divided the population size stop to increase and it starts to decrease when the cells start to die due to too much time spent trying to repair the DNA.

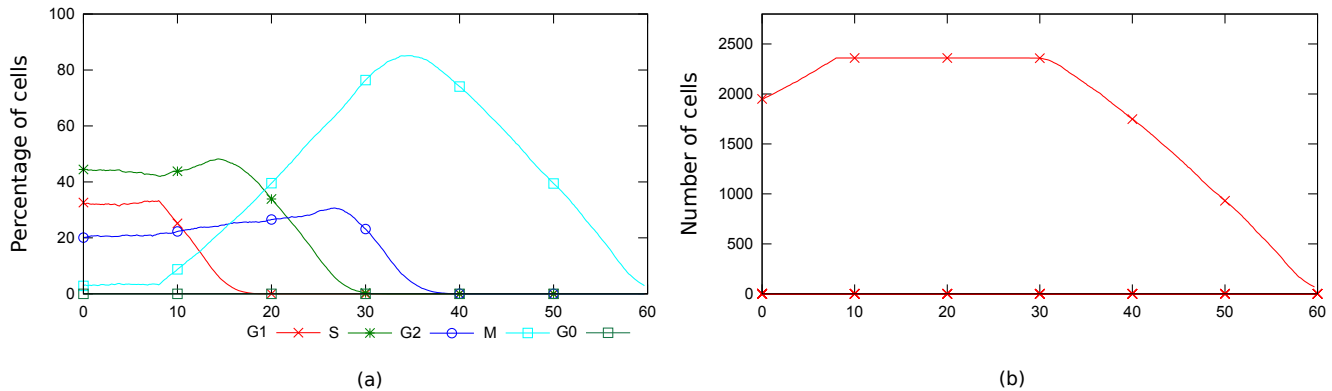


Figure 7: Results of the activation of the intra-mitotic checkpoint. (a) Evolution of the population in each phase. Cells start to accumulate in Mitosis once the nocodazole adjunction occurs. (b) Evolution of the population size. The cells stop to proliferate whereas they accumulate in Mitosis. They start to die when they have spent too much time trying to divide.

programming or other evolutionary strategy. The scenario could be represented as a sequence of parametric actions of the environment and therefore, in specific conditions, a goal could be specified and the best scenario, fitting with this goal, could emerge. The analysis of this scenario could help the biologists to determine qualitatively which regulatory pathway has to be targeted to avoid the uncontrolled proliferation.

The next step of this generic representation of cell cycle is to model the checkpoint permissiveness. If a checkpoint is permissive, cells will pass through the transition whereas the required preconditions are not fulfilled. For instance, the second experiment of this paper will be a test for this module. The G1/S checkpoint being permissive, the cells will pass through it rather than repairing their DNA. We want to express it as a cell belief, using a membership function, as in fuzzy logic, and therefore the combinations of the different

conditions will be aggregated to let the checkpoint activate itself or not. This framework will allow to bring the internal and external perception of the cells to the same level. This representation will simplify the model giving a unique probability of transition for a given checkpoint, and therefore, the FSM presented in section 2 will be transformed in a Markov model.

The different modeling steps followed reduce the side-effect of the cell-environment interactions. The comparison between constrained and unconstrained *in-silico* simulations should allow the quantification of the impact of the environmental constraints. Therefore, the simplified environment will shortly be extended to a 2-D continuous environment and, finally, to a 3-D continuous environment. The final aim of simulating the spatial organization of multicellular tumor spheroids will thus be within reach. As an intermediate step, all the 2-D monolayer classical experiments done

in-vitro will be reproduced *in-silico*. This step will evaluate the response and the influence of the physical model by comparing the *in-vitro* experiments with the results of the simulation with the proposed simplified environment.

Precisely, this 2-D prototype is currently under validation by evaluating the convergence of *in-vitro* experiments and *in-silico* simulation with specific scenarii. For example, we will use the following validation experiments: cell cycle synchronization through a lack of environmental factors (arrest in G0); cell cycle synchronization using a procedure known as double thymidine block (arrest at G1/S) etc. All these experiments will be evaluated with different environments to quantify their impacts.

Conclusion

In this paper the strengths of the checkpoint orientated modeling are highlighted. This approach allows to easily synchronize the cells thanks to environmental interactions. Our cell cycle model gives consistent results for each checkpoint that we try to analyze the response. The probabilistic modeling is an original approach to express specific features of the cell cycle. In this work, it has been shown how a whole population of cells could be synchronized. Nevertheless, this work actually needs to be compared with *in-vitro* experimental results. This multi-disciplinary approach will allow to map some experimental data to parameters value.

References

- Aguda, B. and Tang, Y. (1999). The kinetic origins of the restriction point in the mammalian cell cycle. *Cell proliferation*, 32(5):321–335.
- Calzone, L., Thieffry, D., Tyson, J., and Novak, B. (2007). Dynamical modeling of syncytial mitotic cycles in *Drosophila* embryos. *Molecular systems biology*, 3(1).
- Chavoya, A. and Duthen, Y. (2008). A cell pattern generation model based on an extended artificial regulatory network. *Biosystems*, 94(1-2):95–101.
- Chen, K., Calzone, L., Csikasz-Nagy, A., Cross, F., Novak, B., and Tyson, J. (2004). Integrative analysis of cell cycle control in budding yeast. *Molecular Biology of the Cell*, 15(8):3841.
- Cussat-Blanc, S., Pascalie, J., Luga, H., and Duthen, Y. (2010). Morphogen positioning by the means of a hydrodynamic engine. In *Artificial Life XII*. MIT Press, Cambridge, MA.
- Doursat, R. (2006). The growing canvas of biological development: Multiscale pattern generation on an expanding lattice of gene regulatory networks. *InterJournal: Complex Systems*, 1809.
- Drasdo, D. and Loeffler, M. (2001). Individual-based models to growth and folding in one-layered tissues: intestinal crypts and early development. *Nonlinear Analysis-Theory Methods and Applications*, 47(1):245–256.
- Elledge, S. (1996). Cell cycle checkpoints: preventing an identity crisis. *Science*, 274(5293):1664.
- Galle, J., Loeffler, M., and Drasdo, D. (2005). Modeling the effect of deregulated proliferation and apoptosis on the growth dynamics of epithelial cell populations in vitro. *Biophysical journal*, 88(1):62–75.
- Gerlee, P. and Anderson, A. (2007). An evolutionary hybrid cellular automaton model of solid tumour growth. *Journal of theoretical biology*, 246(4):583–603.
- Hoehme, S. and Drasdo, D. (2010). A cell-based simulation software for multicellular systems. *Bioinformatics*.
- Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467.
- Loeffler, M. and Roeder, I. (2004). Conceptual models to understand tissue stem cell organization. *Current opinion in hematology*, 11(2):81.
- Moreira, J. and Deutsch, A. (2002). Cellular automation models of tumor development: a critical review. *Advances in Complex Systems*, 5(2/3):247–268.
- Novak, B., Pataki, Z., Ciliberto, A., and Tyson, J. (2001). Mathematical model of the cell division cycle of fission yeast. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 11:277.
- Novak, B. and Tyson, J. (1993). Numerical analysis of a comprehensive model of M-phase control in *Xenopus* oocyte extracts and intact embryos. *Journal of Cell Science*, 106(4):1153–1168.
- Novak, B. and Tyson, J. (2004). A model for restriction point control of the mammalian cell cycle. *Journal of theoretical biology*, 230(4):563–579.
- Pascalie, J., Lobjois, V., Luga, H., Ducommun, B., and Duthen, Y. (2011). A Checkpoint-Orientated Model to Simulate Unconstrained Proliferation of Cells (regular paper). In *European Conference on Artificial Life (ECAL), Paris, 09/08/2011-12/08/2011*, pages 630–637, <http://www.mitpress.mit.edu/>. The MIT Press.
- Pascalie, J., Lobjois, V., Luga, H., Ducommun, B., and Duthen, Y. (2012). Checkpoint-Orientated Cell Cycle Modeling - Critical Role for cell age distribution (submitted). In *Gecco 2012*.
- Pomerening, J., Kim, S., and Ferrell Jr, J. (2005). Systems-level dissection of the cell-cycle oscillator: bypassing positive feedback produces damped oscillations. *Cell*, 122(4):565–578.
- Singhania, R., Sramkoski, R., Jacobberger, J., Tyson, J., and Beard, D. (2011). A Hybrid Model of Mammalian Cell Cycle Regulation. *PLoS Computational Biology*, 7(2):835–842.

Computational tests of a thermal cycling strategy to isolate more complex functional nucleic acid motifs from random sequence pools by in vitro selection

Aaron Reba¹, Austin G. Meyer¹, and Jeffrey E. Barrick^{1,2}

¹Department of Chemistry and Biochemistry, ²Center for Systems and Synthetic Biology, Center for Computational Biology and Bioinformatics, Institute for Cellular and Molecular Biology, The University of Texas at Austin
jbarrick@cm.utexas.edu

Abstract

The dual information-function nature of nucleic acids has been exploited in the laboratory to isolate novel receptors and catalysts from random DNA and RNA sequences by cycles of in vitro selection and amplification. This strategy is particularly effective because, unlike polypeptides with random amino acid sequences, nucleic acids with random base sequences are often capable of stably folding into defined three-dimensional structures. However, the pervasive base-pairing potential of nucleic acids is also known to lead to kinetic traps in their folding landscapes. That is, the same DNA or RNA sequence can often adopt alternative base-paired structures that are local energy minima, and these folds may interconvert very slowly. We have used simulations with nucleic acid folding algorithms to evaluate the effect of misfolding on in vitro selection experiments. We demonstrate that kinetic traps can prevent the recovery of novel families of complex functional motifs by two mechanisms. First, misfolding can lead to the stochastic loss of unique sequences in the first round of selection. Second, frequent misfolding can reduce the average activity of multiple copies of a sequence to such an extent that it will be outcompeted after multiple rounds of selection. In these simulations, adding thermal cycling to sample multiple folds of one sequence during a selection for a self-modifying catalytic activity can improve the recovery of rare examples of more complex structures. Although newly isolated sequences may fold poorly, they can represent footholds in sequence space that can be improved to reliably fold after a few mutations. Thus, it is plausible that thermal cycling by day-night cycles or other mechanisms on the primordial earth may have been important for the evolution of the first RNA catalysts, and a fold sampling strategy might be used to search for more effective nucleic acid catalysts in the laboratory today.

Introduction

Catalytic and ligand-binding nucleic acids have diverse applications in gene therapy, synthetic biology, and biotechnology (Breaker, 2004). However, most functional nucleic acids that have been designed or isolated in the laboratory by in vitro selection or directed evolution do not perform adequately for advanced gene control and biosensor applications, especially under in vivo conditions. In some cases, it is clear that there are no inherent chemical limitations that preclude identifying better nucleic acid catalysts and receptors: biological nucleic acid families exist that perform identical functions more effectively. For example, natural

metabolite-binding RNA aptamers from riboswitches (Barrick and Breaker, 2007) far outclass their laboratory counterparts that bind the same molecules (Burgstaller and Famulok, 1994; Kiga, et al. 1998). Riboswitches have larger and more complex structures that achieve greater binding affinities and discriminate better against closely related compounds.

Enriching new functional nucleic acid molecules from random sequence pools by rounds of in vitro selection and enzymatic amplification tends to recover the shortest motifs with the required activity. Thus, the simple consensus motif of the 8-17 deoxyribozyme family usually dominates DNA pools selected for RNA-cleavage activity (Santoro and Joyce, 1997; Cruz, et al. 2004), and RNA pools selected for self-cleavage teem with small hammerhead ribozymes (Salehi-Ashtiani and Szostak, 2001). These simple motifs may obscure more complex families of structures that exist in the same random sequence pool simply because they are numerically so much more likely to be specified by arbitrary nucleotide sequences. In fact, additional families of deoxyribozymes and ribozymes have been found in these two cases by efforts to detect rare variants (Tang and Breaker, 2000; Lam, et al. 2011).

An increase in complexity of five additional base pairs or five new invariant base constraints in the consensus structure of an RNA family has the potential to increase the optimal binding affinity or catalytic rate of a functional RNA family by an order of magnitude (Carothers, et al., 2004). When only a few "winning" sequences are sampled at the end of an in vitro selection experiment, rare sequences—potentially representing more complex structural motifs—may not be recovered solely due to this "tragedy of the commons" (Wilson and Szostak, 1999). However, it is possible that other factors also limit the recovery of more sophisticated functional nucleic acid from random sequence pools.

Many natural RNA molecules from biology are known to adopt multiple, stable base-paired structures. Sometimes this structural degeneracy is necessary for function, as in the case of riboswitches, where different conformations in an "expression platform" sequence are triggered by ligand binding to an "aptamer" domain (Barrick and Breaker, 2007). Alternative conformations have also been harnessed to engineer allosteric ribozymes in the laboratory that use ligand binding to restructure and thereby modulate the activity of the catalytic domain so that they can act as gene control elements or biosensors (Breaker, 2002; Win and Smolke, 2007).

More often structural degeneracy is an undesirable trait,

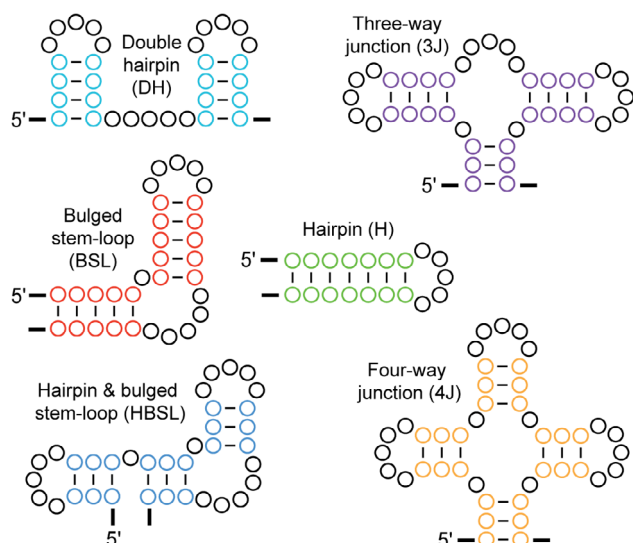


Figure 1. Nucleic acid structure targets

Secondary structure models representing functional RNA families. Circles designate where nucleotides must be present. Each target structure specifies an arrangement of one to four helices, each with three to seven base pairs (colored), and bases with no pairing constraints (black). The functional score of a folded sequence for a particular target is the number of consensus base pairs matched in the structure.

and molecules that misfold into off-pathway conformations are functionally "dead". RNAs from biological systems that form complex structures often require protein chaperones to overcome these kinetic traps and reliably achieve their functional folds (Herschlag, 1995). For example, the group I self-splicing intron misfolds in vitro into a long-lived intermediate that can only refold to reach the native state on timescales of many minutes (Russell, et al. 2006). Even "winning" nucleic acid sequences from laboratory selection experiments often have problems folding reliably. For example, a sizeable subpopulation of misfolded molecules was found to contribute to incomplete self-cleavage of RNA-cleaving deoxyribozymes (Carrigan, et al. 2004). Misfolding is more likely to be a problem for large structures, thus it may prevent the recovery of the most interesting sequences present in random nucleic acid pools.

The nucleotide sequences flanking a functional RNA can profoundly affect its activity, presumably because certain sequence contexts can promote or inhibit correct folding of the functional motif. In one case, the apparent in vitro binding affinity of the same riboswitch aptamer changed by 20-fold depending on the surrounding sequence context (Winkler, et al. 2002). In another, addition of random flanking sequences to a family of laboratory evolved RNA ligases caused their apparent rate constants to vary over more than two orders of magnitude (Sabeti, et al. 1997). It is possible that many functional motifs in random sequence libraries are present in unfavorable contexts that cause them to misfold. Perhaps for this reason, isoleucine aptamers were actually more abundant in 50- and 70-base random regions than in a longer library with 90 random bases, which is a counterintuitive result from a pure probability standpoint (Legiewicz, et al. 2005).

Secondary structure prediction programs have been used as model systems for examining evolutionary trajectories and the overall distribution of function in nucleic acid sequence space (Ancel and Fontana, 2000; Stadler, et al. 2001; Cowperthwaite, 2006). These studies typically use the predicted minimum free energy structures of RNA sequences to construct genotype-to-phenotype maps and an associated fitness landscape. Other studies have calculated the theoretical representation of motifs in random sequence pools (Sabeti, et al. 1997) and used secondary structure prediction algorithms to estimate how often a motif will be correctly folded in the context of flanking sequences (Knight, et al. 2005). These approaches consider the effects of sequence probability and thermodynamic stability on in vitro selection, but they do not take into account what folds are actually accessible and likely to be sampled by each sequence when misfolding into kinetic traps is possible.

Knowledge of protein folding principles has been used to engineer improved strategies for the directed evolution of proteins (Voigt, et al. 2000; Bloom, et al. 2006). Here, we computationally evaluate a potential strategy for recovering more complex nucleic acid structures from in vitro selection experiments based on properties of their folding landscapes. First, we examine how misfolding can impact the recovery of rare motifs from in vitro selection experiments using kinetic simulations of RNA folding. Then, we show that fold sampling by thermal cycling during selection can theoretically enable the recovery of new classes of rare self-modifying ribozymes with more sophisticated structures.

Results

Kinetic folding model of functional nucleic acids

To model how nucleic acid misfolding impacts in vitro selection, we require (1) a method for scoring the functional capacity of a particular nucleic acid structure and (2) an algorithm for simulating the kinetics of the folding process. In general, larger and more complex RNA structures are necessary (but not sufficient) to achieve better functional characteristics. On the basis of aptamer and ribozyme families that have been optimized by re-selection, an empirical rule has been proposed that requiring 10 additional bits of information to be specified (equivalent to 5 new invariant bases or base pairs) in the consensus structure for a functional RNA family has the potential to improve reaction rate (k_{cat}) or binding affinity (K_d) 10-fold (Carothers, et al., 2004). It is reasonable to assume that additional functional information in a structural motif could instead improve specificity against noncognate ligands or reduce requirements for high-concentration divalent metal ion cofactors, two other desirable traits for nucleic acids used to make advanced biosensors or deployed in vivo.

To score functional capacity, we test RNA folds against an arbitrary set of target structures consisting of different arrangements of helical elements with three to seven base pairs (Fig. 1). These structures range from very simple (a hairpin with one helical element) to more complex (a four-way junction of helical elements). Folded sequences receive a score equal to how many base pairs they match in a target structure. For simplicity, we constrain the linker and loop regions between helical elements to fixed lengths. Additional

pairs formed by bases in these connectors count neither for nor against the score. Folded sequences are considered functional members of a target family if they match a certain number of the possible base pairs in the consensus pattern.

In consensus secondary structure models of real functional DNAs and RNAs, the identities of bases at certain positions are invariant (e.g., always an adenine) or are limited to a subset of the four nucleobases (e.g., always a purine). Requirements for specific bases may exist at motif positions that contribute to non-canonical base interactions, tertiary structure contacts, ligand binding, or catalysis (Barrick and Breaker, 2007). These bases are often key determinants of function in an RNA family. However, we do not consider invariant base requirements in our target RNA structures for two reasons. First, secondary structure prediction algorithms used to model folding only calculate the energies of structures with canonical Watson-Crick and G-U base pairs. Second, we are testing hypotheses about misfolding in our model, rather than attempting to calculate the abundance of known functional RNA structures in sequence space, as has been the aim of other studies (Knight, et al. 2005). Adding additional constraints on the identities of bases in our target structures would reduce the chances of sampling "functional" sequences, but because these constraints would be arbitrary, it would probably not enhance the realism of the model.

To simulate RNA folding trajectories, we used version 1.3 of the software program KINFOLD (Flamm, et al. 2000), distributed with version 2.02 of the VIENNA RNA SECONDARY STRUCTURE PACKAGE (Hofacker, et al. 1994). Given an input sequence, KINFOLD traverses the energy landscape of possible secondary structures by the stepwise formation or dissociation of base pairs. We overrode the default behavior which stopped a folding trajectory when the global minimum energy structure was reached. This choice allowed thermal fluctuations to continue for the entirety of the specified folding period. To reduce the number of structures that had to be scored for function, we also used the option to only output states encountered during folding trajectories that were local energy minima. As these structures are typically maximally base paired, this set should generally contain the best possible matches to the target structures of all folds traversed. All simulations used the default folding temperature of 37°C.

We further assumed that a fold with a functional score in our model system possesses an efficient catalytic activity (such as self-cleavage or self-ligation) that needs to be triggered only once to allow it to "pass" an in vitro selection step. Therefore, as a first approximation, we assigned a sequence the maximum score achieved by any base-paired conformation that it adopted during a folding run, without taking into account the overall residence time in different structures. Nucleic acid folding in selection experiments is typically initiated by adding divalent cations (such as Mg^{2+}) to a sample. These positively charged metal ions coordinate the negatively charged phosphate backbone and enable collapse to a compact tertiary structure. We simulated this experimental treatment by beginning each computational folding trajectory with a fully extended strand containing no base pairs.

As an example, we show the results of the entire folding and scoring procedure applied to one sequence (Fig. 2). This sequence achieved a perfect match to the bulged-stem loop structure in 183 of 1000 folding trials (18.3%) lasting 100

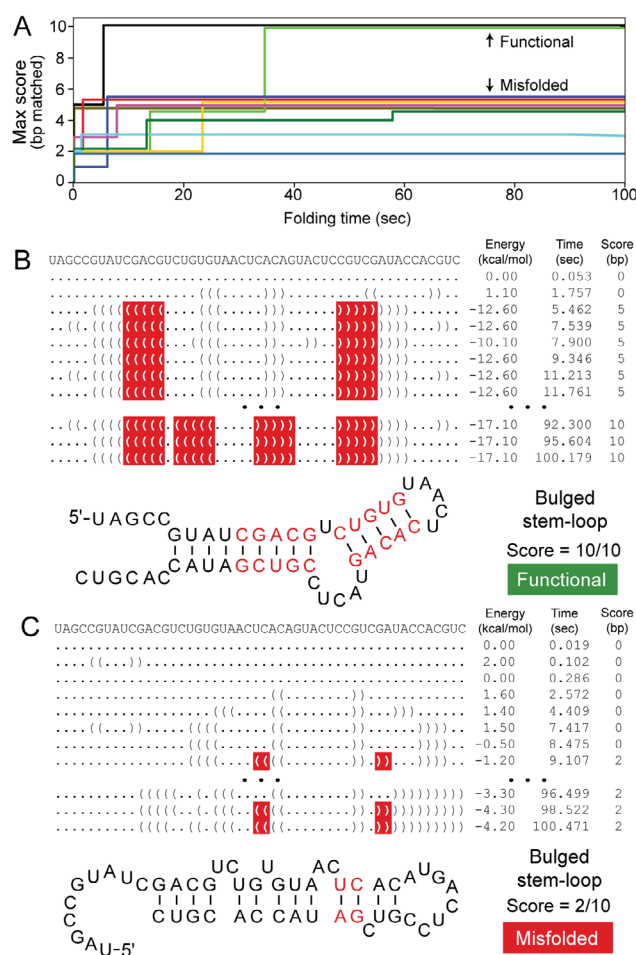


Figure 2. Example of a poorly folding RNA sequence

(A) Ten folding trajectories initiated from an extended strand for the same example sequence. The maximum functional score for the bulged stem-loop target structure encountered up to the given time is shown. Two trials reached the maximum score of 10 within 100 seconds and were judged functional. Only integer scores are possible, so some lines have been offset slightly to avoid overlap. (B, C) Two example folding trajectories. Structures encountered during the initial and final portions of the kinetic simulations are shown. Dots represent unpaired bases and nested sets of parentheses indicate that two bases are paired. The final folded structures are drawn in each case with base pairs that contribute to the score highlighted in red. The first trajectory in (B) achieves a perfect match. The second trajectory in (C) becomes kinetically trapped in an alternative, misfolded structure that is not functional. Only structures that are local energy minima encountered during the simulation of the folding trajectory are shown and scored.

seconds. In the other trials, it collapsed into alternative secondary structures that were unable to rearrange into the functional target structure during the remainder of the folding trajectory, even though it is thermodynamically far more stable. Thus, this poorly folding sequence demonstrates that our computational model recapitulates the key features of real RNA folding landscapes where kinetic traps can prevent many

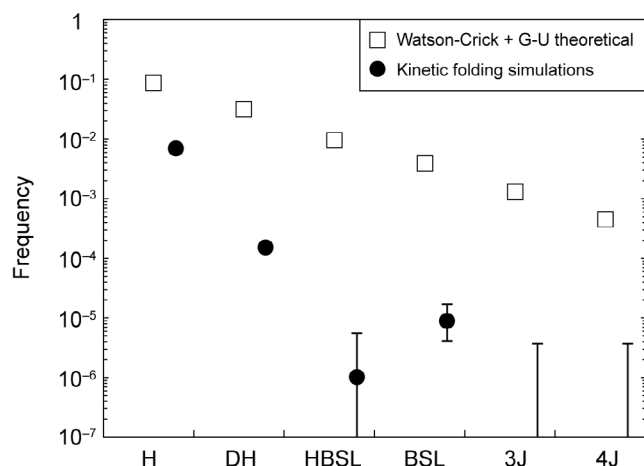


Figure 3. Simulated and theoretical maximum frequencies of random sequences folding to match target structures

Chances that a random sequence could possibly fold to perfectly match each target family were calculated according to the formula in the text (open squares). Sequences actually folded to perfectly match the target structures during kinetic folding simulations at much lower frequencies (filled circles). Error bars are Clopper-Pearson (exact) binomial 95% confidence intervals. Structure abbreviations are given in the Fig. 1 legend. No perfect matches were observed for the 3J or 4J families in the 1,000,000 total sequences that were tested.

individual molecules of a potentially functional sequence from reaching an active conformation.

Misfolding reduces the effective library size

A typical experimental design when trying to isolate new functional RNA families is to begin in vitro selection with a collection of as many random sequences as possible. Each sequence in this "pool" or "library" is often represented just once. Therefore, if a molecule fails to reach the active conformation during the first round of selection, due to becoming energetically trapped in an alternative base-paired structure, that unique sequence will be forever lost from the experimental population. Molecules that do make it through this initial selection step are amplified into hundreds to thousands of copies each before the next selection cycle. Thus, they are generally not expected to be subject to the same dangers of stochastic loss in later rounds. Given the importance of the first round in determining how many unique sequences are interrogated for function, we first investigated the impact of misfolding on this step of in vitro selection.

To what degree does misfolding prevent RNA sequences that are theoretically capable of adopting a functional fold, with all the required base pairs, from achieving this structure? We can calculate p , the probability that a random sequence is capable of perfectly matching one of our target structures, as

$$p = (L - l + 1) b^n$$

where L is the total length of each random sequence, l is the

length of the functional target motif, n is the number of base pairs in the target structure, and b is the chance of a randomly selected base being able to pair with an existing base at a specific position elsewhere in the target structure. For random sequences with equal probabilities of each nucleotide, the parameter b is 0.375 when allowing Watson-Crick or G-U base pairs. Note that this calculation does not guarantee that this structure is the most thermodynamically favorable conformation for a sequence, only that it could adopt this fold. Thus, these calculations yield a theoretical upper limit on the frequency of functional sequences matching a target in a random pool.

To evaluate the effects of misfolding, we simulated a single 10-second folding trajectory for each of 1,000,000 random sequences of length 100. Under the assumption that we are selecting for self-modifying ribozymes, as explained above, we assigned each sequence the best score achieved by any structure encountered in its folding trajectory. We classified the folding trial as resulting in catalytic activity, which would enable that sequence to survive the first round of in vitro selection, if it had a perfect score that matched all base pairs. This procedure enabled us to compare the actual frequency of sequences that were able to kinetically fold to match each target structure to the theoretical calculations (Fig. 3).

The theoretical probability of finding each target structure in a random sequence library decreases nearly uniformly as additional base pairs are added. The chance of finding a random sequence that adopts each target structure in a simulated folding trajectory declines more rapidly as the structures become more complex. The ratio of the two frequencies represents the effective reduction in library size for that structure, i.e., how many times as many sequences as expected would have to be interrogated by in vitro selection to recover a properly folded example of that motif.

The effective library size reduction due to misfolding is 13, 200, 9400, and 430 for the four motifs where perfect matches were found, ordered by the number of base pairs in each target structure. These reductions demonstrate that misfolding disproportionately decreases the likelihood of more complex structures surviving the first round of selection. Thus, not only are more complex RNA structures less likely to be specified by random sequences in the first place, but their inability to reliably fold into the active conformation may compound the combinatorial difficulty of recovering them from a selection.

There is also substantial family-to-family variation in these results. While the effective library size reduction is roughly the same for the double hairpin (DH) and bulged stem-loop (BSL) structures, the effect of misfolding is more than 20-fold greater for the intermediate hairpin & bulged stem-loop (HBSL) structure. This result suggests that the requirement for three helical elements in this motif, compared to two in the others, may have a bigger influence on complexity and misfolding than the relative number of base pairs.

Misfolding reduces the average activity of sequences

In later rounds of in vitro selection, each surviving nucleic acid sequence has been amplified so that it is present many times in the pool. At this stage, the hazard of stochastic loss due to sampling is not as great, as it was in the first round of selection. However, misfolding may still decrease the apparent activity of a sequence if some fraction of a set of

identical molecules becomes kinetically trapped in a "dead" structure upon folding and do not react. If misfolding is prevalent, it can potentially disfavor certain sequences to such an extent that they will not be recovered by in vitro selection. If a structural family is so rare that it is only represented by a few sequences in the entire random pool, all of these may be so poorly folding that they are outcompeted by well-folding examples of more common families at each round of selection. Depending on the extent of misfolding and selection conditions, this can be true even if the rare structure has better performance characteristics when it is correctly folded.

In our simulations, the average activity of many copies of one RNA sequence is equal to the fraction of folding trajectories that sample a conformation with a functional score above some threshold. To more closely examine how kinetic traps can affect average activity and winners over multiple rounds of selection, we simulated 10,000 folding trajectories lasting 10 seconds for each of 10,000 random sequences of length 100. Since we were limited to fewer total sequences, we scored as functional not only perfectly scoring matches, but also structures with as many as two missing base pairs. To summarize the results we plotted the folding characteristics of the ensembles of structures achieved by each sequence in separate folding trials as the cumulative distribution of the number of different sequences that achieved a functional score in some fraction of their folding trajectories (Fig. 4).

We found that even the best-folded examples of each family in this limited sample of sequences encountered kinetic traps such that they were not able to reach the functional structure a large fraction of the time. To get an idea of the impact this might have on an experiment, assume that selection takes place under permissive conditions where all correctly folded sequences react to completion. Then, the relative enrichment of the best examples of each family (y-intercepts in Fig. 4) as they replace fully nonfunctional sequences would be equal to their relative chances of folding into an active structure. Under these conditions, the three most complex structures (HBSL, 3J, and 4J) would be outpaced by the simpler structures by at least 10-fold in every round of selection.

It is possible that more complex structures might have improved performance characteristics, such as tighter binding or a higher catalytic rate. To some extent, this might mitigate the advantage of the smaller structures. With suitably stringent selection conditions, which capture ligand binding with very low off rates for a receptor or very rapid initial kinetics for a catalyst, increased function can almost directly translate into better recovery of a sequence. Under these conditions, a representative of the 3-way junction (3J) motif family would still need almost 100-times the activity of a representative of the double hairpin (DH) structure to compete on equal footing for enrichment and recovery, for example. However, this high stringency could also put each structure at risk for stochastic loss again, as in the first round of selection. In the example, only one in 10^3 molecules with an active sequence would be recovered after a selection step that was stringent enough to yield equal enrichment of the two motifs.

The advantages of smaller structures in the average effective activity they realize compared to larger structures will occur in each of the 8 to 15 total rounds of in vitro selection and amplification in a typical experiment. Thus, the

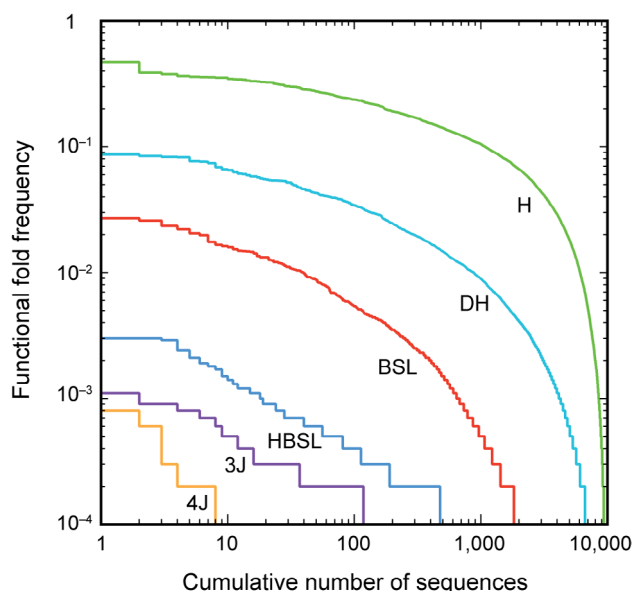


Figure 4. Reduction in average activity due to misfolding

The cumulative numbers of sequences that folded to achieve a functional conformation in a given fraction of simulated secondary structure folding trajectories are shown. For these simulations the score judged to be "functional" for each family was the maximum possible score minus two. Structure abbreviations are given in the Fig. 1 legend.

large reduction in the effective activity of the more complex structures due to misfolding would be a substantial obstacle to recovering a rich collection of functional RNA families.

Thermal cycling relaxes selection against misfolding

Because we are simulating selection for a self-modifying catalytic function, a molecule only needs to reach the active state once to pass selection. We reasoned that giving each RNA molecule multiple opportunities to refold within a single round of selection would relax selection against poorly folding sequences. This effect could improve the recovery of complex nucleic acid families, which will be rarer in a random pool, and therefore more likely to be lost due to misfolding during the first round of selection or to be disfavored relative to simpler structures by reduced activity in later rounds.

Thermal cycling would be a simple way to implement fold sampling. We simulated thermal cycling by splitting up a folding interval of constant total length (1,000 sec) into one, ten, or one hundred separate folding cycles and examining the maximum activity achieved by 100,000 sequences of length 100 during this time (Fig. 5A). We simulated each thermal cycling step by restarting a new folding trajectory from an extended conformation with no secondary structure. This procedure is equivalent to heating achieving perfect unfolding followed by an instantaneous return to normal temperature.

Overall, thermal cycling treatments increased the functional scores achieved by the random sequences that were tested, as expected (Fig. 5B). We quantified this improvement in two ways. First, we asked how the average score that sequences achieved for each target family changed (Fig. 5C). We found that the ten folding cycle treatment improved the average

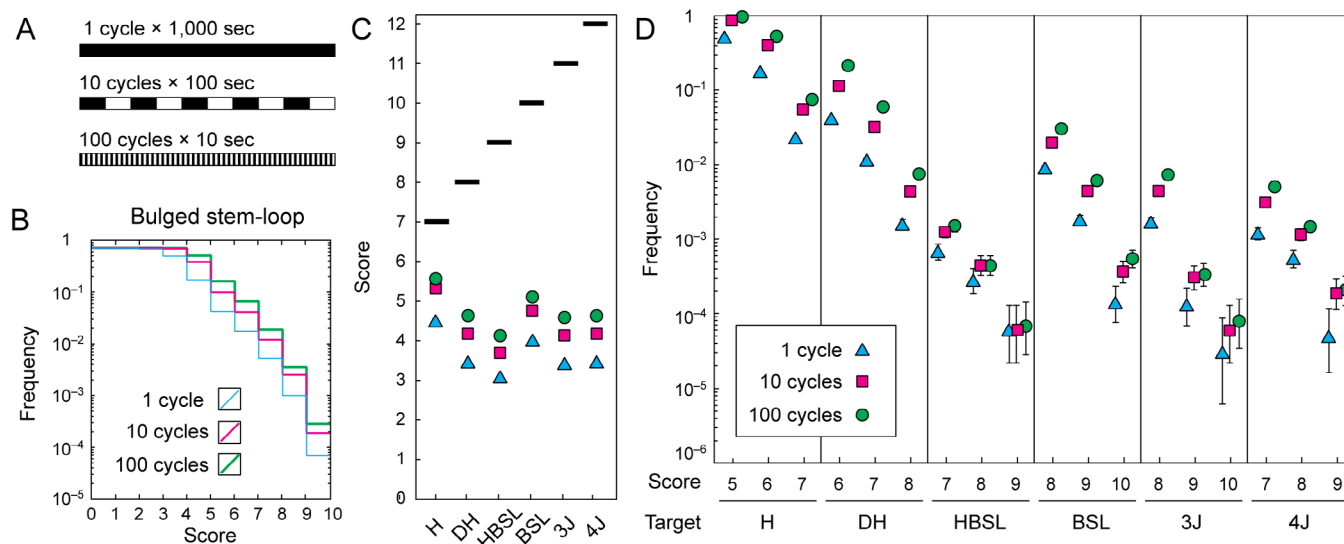


Figure 5. Fold sampling by thermal cycling increases the chances that random sequences achieve active structures

(A) 10,000 sequences were folded for a total of 1,000 seconds each, divided into 1, 10, or 100 folding cycles as illustrated. (B) The cumulative distributions of the maximum scores achieved by each of these sequences for the bulged stem-loop target structure in each folding treatment. (C) The average score achieved for each structure in each folding treatment. Black bars are the maximum possible scores. (D) The frequency of sequences scoring above the given high score cutoffs for each target structure. Error bars are exact binomial (Clopper-Pearson) 95% confidence intervals to show the error due to sparsely sampling low numbers of active structures. They are hidden behind symbols in many cases. Structure abbreviations are given in the Fig. 1 legend.

score for each family by 0.61 to 0.82 points (mean: 0.72). The one-hundred folding cycle treatment improved the average score by 1.06 to 1.17 points (mean: 1.12). This result roughly translates into the 100-fold treatment matching one additional base pair in the target compared to the 1-fold treatment.

Since truly functional sequences are better represented by the high-scoring tail of this distribution than by its average, we also asked how often sequences scoring above a threshold value were found in the multiple folding treatments (Fig. 5D). For the more complex sequence families the average increase in the frequency of sequences that achieved a high score (defined as within three of the maximum where activity was observed, for example 8, 9, or 10 for the three-way junction) was roughly constant across all structures. We found that, on average, the effective library size for those structures after cycling would be increased approximately 2.4-fold for 10 cycles and 3.4-fold for 100 cycles, excluding the hairpin structure. Again, this is roughly equal to increasing the effective library size for finding complex structures by one consensus base pair. Despite the larger apparent hazards of complex structures misfolding in the first round of selection described above, there was no apparent effect of the structural complexity of the target motif on the improvement in effective library size from thermal cycling.

Multiple folding by thermal cycling could also mitigate the reduction in average activity due to misfolding at each round of selection. Due to computational costs, we were unable to repeat the longer 100-second and 1,000-second folding trials multiple times for each sequence to fairly judge the relative success of sequences under these conditions. To estimate the possible magnitude of this effect, we again consider just the best-folding examples from the experiment where each sequence was folded 10,000 different times for 10 seconds.

Here every ten cycles of re-folding theoretically increases the fraction of molecules that achieve the active structure by about 10-fold, until it becomes close to one. Therefore, the 100-cycle folding treatment should bring the three smallest structures up to parity in terms of effective activity per round of selection. Therefore, this consequence of thermal cycling might have an even more substantial impact on the eventual recovery of complex structural motifs.

Poorly folding sequences may be evolutionary footholds

Thus far we have provided evidence that thermal cycling can theoretically increase the number of functional structures recovered by in vitro selection by relaxing selection against misfolding. However, a majority of the new sequences recovered by this procedure would be poorly folding: by definition their "function" is conditional on being thermally cycled many times. Our primary interest is to recover rare examples of larger, more complex structures that would not be found otherwise. Due to sparse sampling of sequence space, recovering even a handful of poorly folding examples of new structural motifs might establish evolutionary "footholds" in the sequence-structure landscape. It is possible that some of these beachhead sequences could be readily optimized to well-folded examples of a structural family in only a few additional mutational steps. Alternatively, most sequences newly recovered by the thermal cycling strategy might be pathologically poor folders that are not near any well-folding examples in sequence space, causing them to remain trapped in a poorly functioning limbo that cannot be improved.

To determine whether it was plausible that this procedure would find useful evolutionary footholds, we surveyed the local mutational landscape of two random poorly folding

examples of structural classes found in the thermal cycling simulations. We measured the fraction of the time that the original sequence, and all of its one-mutation neighbors, reached functional folds in 10-second trajectories. Then, we took the best one-mutant neighbor and looked at its one-mutant neighborhood. This procedure is roughly analogous to a re-selection experiment from a doped pool, where a single master sequence (that was recovered by the original selection) is resynthesized with a probability of error at each position, and selection is used to find variants with optimized function.

We found that the original example of the three-way junction (3J) class folded to within one base pair of a perfect score in 10/10000 trials. Its best single-mutant neighbor folded to this level of function in 6/1000 trials after a change in the flanking sequences. However, there are 300 possible single-step mutations for this sequence of length 100, and the observed increase is not statistically significant when corrected for multiple testing (Bonferroni corrected Fisher's exact test, $p = 0.60$). A second single mutant of this sequence improved to be capable of a perfect score by changing a base in the outermost pair of the target, but it only reached this structure in 1/1000 trials.

An example of the bulged stem-loop (BSL) structure initially folded to a perfect match in 137/10000 folding trajectories. The best single mutant reached this score in 35/1000 trials and the best further single mutant of this sequence was active in 46/1000 trials. The first of these increases is highly significant (Bonferroni corrected Fisher's exact test, $p = 0.0016$) and resulted from changing a G-U base pair to a more thermodynamically stable G-C base pair. In this second case, we see evidence that some of the poorly folded structures isolated by the thermal cycling procedure can be improved with very few mutations. Since changes to more realistic RNA structures often require simultaneous changes to covarying base pairs, it may be that our step-wise procedure is not as effective as a real re-selection experiment where substantially more variation can be introduced and filtered.

Discussion

We have used kinetic secondary structure folding simulations to examine how misfolding into kinetic traps can hamper the recovery of novel families of functional nucleic acids by in vitro selection. We showed that misfolding could reduce both the effective library size, by preventing sequences from surviving the first round of selection, and lower the effective activity of sequences in later rounds of selection such that they are outcompeted. Both of these processes bias selection against the recovery of larger motifs, which have the most potential for achieving the greatest function. We found these disadvantages could be mitigated somewhat by using thermal cycling to re-fold molecules multiple times during a selection step, allowing more complex structures to be recovered from experiments. It is our hope that this work serves as a proof-of-principle and prelude for experimentally investigating fold sampling procedures.

Many questions remain about how adjusting the critical parameters in this model will affect the utility of thermal cycling. In particular, we did not systematically examine how the effectiveness of this strategy depended on the duration of

each folding step. We also used rather small target structures, as only 10^5 sequences could be simulated. Much larger functional motifs are usually isolated from real random nucleic acid libraries that contain upwards of 10^{15} sequences. While we expect the general properties of the folding landscape to remain the same for larger structures, the magnitude of many effects could be quite different. Future studies might examine larger structures by using inverse folding algorithms (Hofacker, et al. 1994) to design sequences with minimum free energy structures that are functional, and examining how misfolding affects their recovery.

A number of other folding strategies could also be investigated with this kinetic folding model. First, adding chaperones (Herschlag, 1995) during the selection step could enable sequences to more reliably reach their global energy minimum rather than being stuck in more shallow local minima. This treatment might be implemented by simply assigning sequences to their minimum free energy structure. Second, it is possible that certain types of less extreme temperature fluctuations would allow useful exploration of multiple, folded conformations to achieve function. Third, performing selection at elevated temperatures could disfavor poorly folded sequences and give rise to active pools of molecules where only highly structured, and perhaps more complex, sequences that do not melt remain functional.

Temperature cycling has been previously used in limited contexts in nucleic acid selection experiments. In a notable selection with a very large random region consisting of 220 nucleotides, "the temperature was cycled between 25°C and 37°C to encourage individual RNA molecules to explore alternative conformations" (Bartel and Szostak, 1993). Perhaps not coincidentally, a class of ligase ribozymes from this selection formed the basis for the eventual creation of the exceptionally complex RNA-dependent RNA polymerase ribozyme (Johnston, et al. 2001). Temperature cycling has also been used to fold allosteric ribozymes multiple times in the absence of ligand to prevent selecting ribozymes that probabilistically fold some fraction of the time into a conformation capable of self-cleavage. This bet-hedging behavior allows these "cheaters" to be triggered some fraction of the time that they fold and pass selection despite not recognizing the desired ligand (Soukup, et al. 2000).

We have neglected discussing experimental procedures that introduce new mutations, i.e. "evolution" rather than pure "selection". RNA's penchant for adopting alternative structures is a type of phenotypic noise or plasticity (Ancel and Fontana, 2000). A "lookahead effect" has been modeled where phenotypic misexpression of a genotype can enable evolution to more quickly cross fitness valleys to complex traits under certain circumstances (Whitehead, et al. 2008). RNA folding may have more relevant rates of "phenotypic misexpression" for this effect than transcription or translation errors. It has already been shown experimentally that the ability of one RNA sequence to adopt multiple folds can create nearly neutral single-mutation walks between functional structures (Schultes and Bartel, 2000). Thermal cycling during one of these experiments could bring such valley-crossing events even closer to full neutrality.

Early chemical evolution might have relied on fold cycling to give poorly folded, compositionally mixed, or chemically heterogeneous sequences extra chances to function. One can

imagine the early earth as a very large and slow thermal cyler, with day-night cycles re-folding primordial heteropolymers in the soup repeatedly until the "spark" of hitting an active conformation. Fluctuating or interfacial microenvironments, such as thermal vents or hypersaline pools, might cause fold sampling on more rapid timescales.

Not captured by the current kinetic folding model, but also possibly very important in an early RNA World, is the capacity for interstrand base pairing. At each refolding cycle, molecules have multiple opportunities to interact, not only with themselves to form new structures, but also with different partners in a mixture to form functional conglomerates. Thermal cycling could also drive systems that rely on interstrand base pairing for polymerization or ligation (Johnston, 2001; Lincoln and Joyce, 2009). While in many cases these interactions may lead to inhibition or even the evolution of parasites (Hanczyc and Dorit, 1998), they represent another potential type of phenotypic plasticity that can multiply the functional possibilities of nucleic acids.

Acknowledgments

We thank Andy Ellington, Rick Russell, Jared Ellefson, Alvaro Rodriguez, and Vinicio Reynoso for discussing fold sampling strategies and pointing out useful references. We used the Texas Advanced Computing Center (TACC) for simulations. This work was funded by the Welch Foundation (F-1780), National Institutes of Health (R00GM087550), and the National Science Foundation BEACON Center for the Study of Evolution in Action (DBI-0939454).

References

- Ancel, W. and Fontana, W. (2000). Plasticity, evolvability, and modularity in RNA. *The Journal of Experimental Zoology*, 288: 242-283.
- Barrick, J. E. and Breaker, R. R. (2007). The distributions, mechanisms, and structures of metabolite-binding riboswitches. *Genome Biology*, 8:R239.
- Bartel, D. P. and Szostak, J. W. (1993). Isolation of new ribozymes from a large pool of random sequences. *Science*, 261:1411-1418.
- Breaker, R. R. (2004). Natural and engineered nucleic acids as tools to explore biology. *Nature*, 432, 838-845.
- Bloom, J.D., Labthavikul, S.T., Otey, C.R., and Arnold, F.H. (2006). Protein stability promotes evolvability. *Proceedings of the National Academy of Sciences of the United States of America*, 103:5869-5874.
- Breaker, R. R. (2002). Engineered allosteric ribozymes as biosensor components. *Current Opinion in Biotechnology*, 13:31-39.
- Burgstaller, P. and Famulok, M. (1994). Isolation of RNA aptamers for biological cofactors by in vitro selection. *Angewandte Chemie International Edition*, 33:1084-1087.
- Carrigan, M. A., Ricardo, A., Ang, D. N., and Benner, S. A. (2004). Quantitative analysis of a RNA-cleaving DNA catalyst obtained via in vitro selection. *Biochemistry*, 43:11446-11459.
- Carothers, J. M., Oestreich, S. C., Davis, J. H., and Szostak, J. W. (2004). Informational complexity and functional activity of RNA structures. *Journal of the American Chemical Society*, 126:5130-5137.
- Cowperthwaite, M. C., Bull, J. J., and Meyers, L. A. (2006). From bad to good: Fitness reversals and the ascent of deleterious mutations. *PLoS Computational Biology* 2: e141.
- Cruz, R. P. G., Withers, J. B., and Li, Y. (2004). Dinucleotide junction cleavage versatility of 8-17 deoxyribozyme. *Chemistry & Biology*, 11:57-67.
- Flamm, C., Fontana, W., Hofacker, I., and Schuster, P. (2000). RNA folding kinetics at elementary step resolution. *RNA*, 6:325-338.
- Hanczyc, M. M. and Dorit, R. L. (1998) Experimental evolution of complexity: in vitro emergence of intermolecular ribozyme interactions. *RNA*, 4:268-275.
- Herschlag, D. (1995). RNA chaperones and the RNA folding problem. *Journal of Biological Chemistry*, 270:20871-20874.
- Hofacker, I. L., Fontana, W., Stadler, P. F., Bonhoeffer, L. S., Tacker, M., and Schuster, P. (1994). Fast folding and comparison of RNA secondary structures. *Monatshefte für Chemie*, 125:167-188.
- Johnston, W. K., Unrau, P. J., Lawrence, M. S., Glasner, M. E., and Bartel, D. P. (2001). RNA-catalyzed RNA polymerization: accurate and general RNA-templated primer extension. *Science*, 292:1319-1325.
- Kiga, D., Futamura, Y., Sakamoto, K., and Yokoyama, S. (1998). An RNA aptamer to the xanthine/guanine base with a distinctive mode of purine recognition. *Nucleic Acids Research*, 26:1755-1760.
- Knight, R., de Sterck, H., Markel, R., Smit, S., Oshmyansky, A., and Yarus, M. (2005). Abundance of correctly folded RNA motifs in sequence space, calculated on computational grids. *Nucleic Acids Research*, 33:5924-5935.
- Lam, J. C. F., Kwan, S. O., and Li, Y. (2011) Characterization of non-8-17 sequences uncovers structurally diverse RNA-cleaving deoxyribozymes. *Molecular Biosystems*, 7:2139-2146.
- Legiewicz, M., Lozupone, C., Knight, R., and Yarus, M. (2005). Size, constant sequences, and optimal selection. *RNA*, 11:1701-1709.
- Lincoln, T. and Joyce, G. F. (2009). Self-sustained replication of an RNA enzyme. *Science*, 323:1229-32.
- Russell, R., Das, R., Suh, H., Travers, K. J., Laederach, A., Engelhardt, M. A., and Herschlag, D. (2006). The paradoxical behavior of a highly structured misfolded intermediate in RNA folding. *Journal of Molecular Biology*, 363:531-544.
- Sabeti, P. C., Unrau, P. J., and Bartel, D. P. (1997). Accessing rare activities from random RNA sequences: the importance of the length of molecules in the starting pool. *Chemistry and Biology*, 4:767-774.
- Salehi-Ashtiani, K. and Szostak, J. W. (2001). In vitro evolution suggests multiple origins for the hammerhead ribozyme. *Nature*, 414:82-84.
- Santoro, S. W. and Joyce, G. F. A general purpose RNA-cleaving DNA enzyme. (1997). *Proceedings of the National Academy of Sciences of the United States of America*, 94: 4262-4266.
- Schultes, E. A. and Bartel, D. P. (2000). One sequence, two ribozymes: implications for the emergence of new ribozyme folds. *Science*, 289:448-452.
- Soukup, G. A., Emilsson, G. A., and Breaker, R. R. (2000). Altering molecular recognition of RNA aptamers by allosteric selection. *Journal of Molecular Biology*, 298: 623-632.
- Stadler, B. M., Stadler, P. F., Wagner, G. P., and Fontana, W. (2001). The topology of the possible: formal spaces underlying patterns of evolutionary change. *Journal of Theoretical Biology*, 213:241-274.
- Tang, J. and Breaker, R.R. (2000). Structural diversity of self-cleaving ribozymes. *Proceedings of the National Academy of Sciences of the United States of America*, 97:5784-5789.
- Voigt, C.A., Kauffman, S., and Wang, Z.G. (2000). Rational evolutionary design: the theory of in vitro protein evolution. *Advances in Protein Chemistry*, 55:79-160.
- Whitehead, D. J., Wilke, C. O., Vernazobres, D. and Bornberg-Bauer, E. (2008). The look-ahead effect of phenotypic mutations. *Biology Direct*, 3:18.
- Wilson, D. S. and Szostak, J. W. (1999). In vitro selection of functional nucleic acids. *Annual Review of Biochemistry*, 68:611-647.
- Win, M. N. and Smolke, C. D. (2007). RNA as a versatile and powerful platform for engineering genetic regulatory tools. *Biotechnology & Genetic Engineering Reviews*, 24:311-346.
- Winkler, W. C., Nahvi, A., and Breaker, R. R. (2002). Thiamine derivatives bind messenger RNAs directly to regulate bacterial gene expression. *Nature*, 419:952-956.

Design and construction of a prototype CMY (Cyan-Magenta-Yellow) genetic circuit as a mutational readout device to measure evolutionary stability dynamics and determine design principles for robust synthetic systems

Sean C. Sleight¹ and Herbert M. Sauro¹

¹ University of Washington, Dept. of Bioengineering, Seattle, WA 98195
sleight@uw.edu

Abstract

Synthetic biology is the engineering discipline for constructing novel organisms with functions that do not exist in nature. We recently engineered a prototype CMY genetic circuit that visually produces cyan, magenta, and yellow colors independently and in combination using different inducer molecules. Since the production of each color can be independently controlled, this allows for the production of a spectrum of colors that can be visualized in normal light conditions, and each color can be quantified using fluorescence measurements. We performed an evolution experiment to measure the evolutionary stability dynamics of this prototype CMY genetic circuit in 88 replicate populations of *Escherichia coli*, propagated with all colors turned on. Our results using particular inducer concentrations show that all 88 replicate populations change from a dark, green-brown color to a cyanish color after only 40 generations. In order to visualize the results of this experiment, we washed and concentrated the cells from each population into a different well of a 384-well plate at different evolutionary timepoints. The color change seen visually is confirmed with quantitative data that demonstrates the loss-of-function of both magenta and yellow colors with variation between replicate populations. We sequenced a single clone from four independently evolved populations and all clones have the same loss-of-function deletion mutation between homologous transcriptional terminators that removes the magenta and yellow expression cassettes. This parallel evolution was somewhat expected from results of previous work, but we expect that randomized and re-engineered versions of this circuit without repeats will produce more divergent results due to more stochastic loss-of-function mutations. This prototype CMY circuit serves as a mutational readout device and allows for a colorimetric and quantitative demonstration of evolution in action using synthetic biology.

Introduction

The field of experimental evolution uses controlled experiments for studying evolutionary dynamics over short or long timescales in the laboratory (Elena and Lenski 2003). Evolution experiments in the laboratory normally involve propagating bacteria or other microbes over multiple generations in certain environmental conditions to understand the phenotypic and genetic differences between evolved strains and their progenitors (Riehle, Bennett, and Long 2005; Herring et al. 2006; Schoustra et al. 2006; Sleight and Lenski 2007). These experiments also allow for the study of parallel or divergent evolutionary processes at the genetic and phenotypic levels between replicate evolved populations. Parallel evolution occurs when multiple evolved populations derived from the same ancestor converge on a similar phenotype and is a strong indicator of evolutionary adaptation

(Cooper et al. 2001; Colosimo et al. 2005; Sleight et al. 2008; Meyer et al. 2010; Toprak et al. 2012).

For a number of reasons, synthetic biology offers a powerful system for studying evolution in the laboratory. Synthetic biologists assemble genetic circuits and metabolic pathways from individual genetic “parts” (Knight, 2003; Shetty et al., 2008; Sleight et al., 2010a), normally encoded on plasmids (Elowitz & Leibler, 2000; Gardner et al., 2000; Basu et al., 2005; Levskaya et al., 2005; Entus et al., 2007; Anderson et al., 2007; Sleight et al., 2010b), but also on the chromosome (Tyo et al., 2009; Gibson et al., 2010). Each genetic circuit has a unique metabolic load associated with it, due to the production of foreign proteins, and as a result has a unique fitness. Any cell in the population that removes this metabolic load through a loss-of-function mutation normally has a large fitness increase (unpublished results). Because of this large fitness differential between functional and nonfunctional cells, evolution occurs rapidly due to the functional cells in the population being outcompeted by nonfunctional cells. There are several examples of genetic circuits (You et al., 2004; Balagaddé et al., 2005; Canton et al., 2008; Sleight et al., 2010b) and metabolic pathways (Yoon et al., 2007; Philip et al., 2009; Tyo et al., 2009) that have lost function over evolutionary time. The loss-of-function mutations can easily be determined by comparing the original plasmid sequence with the plasmid extracted from individual evolved clones. One disadvantage to studying evolutionary dynamics in plasmids is that there may be unknown mutations that occur on the host chromosome, but transforming the original and mutant plasmids back into plasmid-less host, allows for determination of the mutant plasmid phenotype (Sleight et al., 2010b). Thus, synthetic biologists are able to engineer different genetic circuits with control over the exact DNA sequence, transform them into a host organism of choice, perform evolution experiments that often occur over short timescales, and determine the exact mutations responsible for evolutionary adaptation.

As genetic circuits get more complex, it becomes increasingly important to understand the evolutionary stability dynamics of large circuits with high metabolic loads. With this goal in mind, we recently engineered a prototype CMY (Cyan-Magenta-Yellow) genetic circuit (**Figure 1**) to study the evolutionary stability dynamics of a three-gene circuit. After various iterations, we found that this circuit expresses colors best using medium-copy plasmids (instead of high-copy which cause instability) and strong ribosomal binding sites. Each color in the circuit can be turned on independently and in combination using different inducer molecules, producing a spectrum of different colors. The color can be

seen visually in normal light conditions when the cells are pelleted, washed, and resuspended in water. Importantly, each color can also be quantified by measuring fluorescence. These colorimetric and quantitative methods allow for a mutational “readout” of circuit function at any evolutionary timepoint. In this study, we independently evolved 88 replicate populations to understand the mutational robustness of this prototype circuit and whether the replicate populations would evolve in parallel or divergently. These results will allow for a better understanding of how to engineer robust synthetic systems.

Materials and Methods

Circuit engineering and use of strains. The prototype CMY circuit (**Figure 1**) was engineered from DNA obtained from the Registry of Standard Biological Parts (partsregistry.org) using the Clontech In-Fusion PCR Cloning Kit, with the specific methods described previously (Sleight et al., 2010a). This circuit was cloned in the pSB3K3 plasmid, a medium copy number plasmid (20-30 plasmids/cell) with a

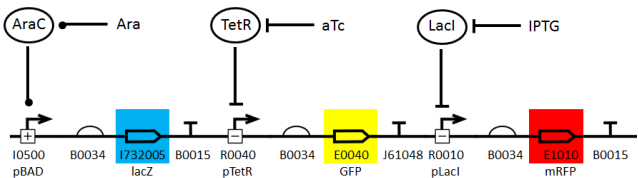


Figure 1. Design and regulation of the prototype CMY genetic circuit. The CMY circuit expresses three proteins (LacZ, GFP, and mRFP) independently and in combination using different inducer molecules. The inducer molecule arabinose (Ara) binds to the AraC protein (encoded on the plasmid) and activates expression of LacZ from the pBAD promoter. The addition of X-gal in the media allows for the visualization of beta-galactosidase (LacZ) expression since LacZ cleaves this molecule to produce a blue color (seen as cyan in normal light conditions if a particular concentration is used). The addition of MUG in the media allows for the quantification of LacZ. The cells turn yellow in normal light conditions when the inducer aTc binds to TetR (encoded on the chromosome) and derepresses expression of GFP from the pTetR promoter. To produce a visual red color, IPTG binds to LacI (encoded on the chromosome) and derepresses expression of mRFP from the pLacI promoter. The genetic symbols (taken from SBOL visual) represent promoters (broken arrows with activation or repression), ribosome binding sites (half ovals), coding sequences (forward arrows), and transcriptional terminators (“T” symbol). Lines ending with a perpendicular line indicate activation and lines ending with a perpendicular line indicate repression. See details in the Materials and Methods section. p15A pMR101-derived replication origin and kanamycin resistance gene (Lutz & Bujard, 1997). This plasmid was transformed into MG1655 Z1 (Sleight et al., 2010b) which constitutively overexpresses LacI and TetR from the chromosome. The AraC protein is expressed from a constitutive promoter on I0500 in the reverse direction, whereas LacZ is expressed from the pBAD promoter on I0500 in the forward direction.

Independent and combinatorial expression of each color on the CMY circuit. The Z1 strain transformed with the CMY plasmid was streaked out from a freezer stock of a culture frozen with 15% glycerol and stored at -80°C. One

colony was grown for 24 hours in 5 mL LB + 50 µg/mL kanamycin in a test tube grown at 37°C shaking at 250 RPM. Eight populations were inoculated (1:1000 dilution) into 5 mL LB + 50 µg/mL kanamycin in test tubes grown at 37°C shaking at 250 RPM and supplemented with different inducers and molecules. Expression of LacZ was induced with 0.02% arabinose from the pBAD (I0500) promoter. X-gal was added to the media to visualize the presence of beta-galactosidase, expressed from the *lacZ* coding sequence. X-gal is cleaved by β-galactosidase yielding galactose and 5-bromo-4-chloro-3-hydroxyindole. The latter is then oxidized into 5,5'-dibromo-4,4'-dichloro-indigo, an insoluble blue product. The molecule 4-Methylumbelliferyl beta-D-galactopyranoside (MUG) dissolved in DMSO was added to the media to quantify the concentration of LacZ using fluorescence measurements (Vidal-Aroca et al. 2006). Green Fluorescent Protein (GFP) was induced with Anhydrotetracycline (aTc) from the pTetR (R0040) promoter. Monomeric Red Fluorescent Protein (mRFP) was induced with Isopropyl-beta-D-thiogalactopyranoside (IPTG) from the pLacI (R0010) promoter. The following inducers and molecules were added to eight controls according to the table below (+ indicates the addition of the inducers and molecules listed in the column and – indicates absence of the same inducers and molecules). Results of this experiment are shown in **Figure 2**.

Inducers and molecules	Arabinose (0.02%), X-gal (20 µg/ml), MUG (2 µg/ml)	aTc (10 µg/mL)	IPTG (1 X 10 ⁻⁴ M)
Control #1	+	-	-
Control #2	-	+	-
Control #3	-	-	+
Control #4	+	+	-
Control #5	-	+	+
Control #6	+	-	+
Control #7	+	+	+
Control #8	-	-	-

Evolution experiment. The Z1 strain transformed with the CMY plasmid was streaked out from a freezer stock of a culture frozen with 15% glycerol and stored at -80°C. One colony was grown for 24 hours in 5 mL LB + 50 µg/mL kanamycin in a test tube grown at 37°C shaking at 250 RPM. Eight-eight identical populations were inoculated from this culture (1:1000 dilution) into 1.5 mL LB + 50 µg/mL kanamycin and supplemented with 0.02% arabinose, 20 µg/ml X-gal, 2 µg/mL MUG, 10 µg/mL aTc, and 1 X 10⁻⁴ M IPTG in an Eppendorf deep 96-well plate sealed with a Thermo Scientific gas permeable membrane for maximum oxygen diffusion. These cultures were grown at 37°C shaking at 250 RPM and propagated every 24 hours achieve about 10 generations per day (log₂ 1000 = 9.97).

Cell density and fluorescence measurements. Every 24 hours, cell density (OD₆₀₀) and fluorescence of evolved populations were measured in a Tecan Infinite M200 Pro fluorescence plate reader. The measurement timepoint chosen was every 24 hours because the rate of change of fluorescent protein expression is close to steady-state. Evolved

populations thus spend about 8-12 hours in lag or exponential phase and the remaining time in stationary phase. For each timepoint, all populations were thoroughly mixed and 200 μ L was transferred into a black, clear-bottom 96-well plate (Costar). Fluorescence was measured for LacZ expression using 360nm excitation/460nm emission wavelengths, GFP using 485 nm excitation/516 nm emission wavelengths, and mRFP using 584 nm excitation/620 nm emission wavelengths. Fluorescence for each color was then divided by OD₆₀₀ to measure the normalized expression (Fluorescence/OD₆₀₀).

Plasmid sequencing. After 40 generations, four evolved populations were streaked out on LB + 50 μ g/mL kanamycin agar plates. One clone from each population was grown in 5 mL LB + 50 μ g/mL kanamycin for 24 hours at 37°C shaking at 250 RPM. Plasmids were extracted using the Qiagen Miniprep Kit and submitted to the Genewiz sequencing facility for sequencing. Purified plasmid DNA was sequenced using VF2/VR primers specific to the pSB3K3 vector (about 100 bp on either side of the circuit) and internal primers specific to the circuit.

Visualizing cell color in controls and evolved populations. The cell color for controls #1-8 were visualized by centrifuging 5 mL test tubes in a Sorvall Legend 23R centrifuge at 3000 RPM for 10 minutes at 4°C, removing the supernatant, washing with 500 μ L of water, centrifuging again at 3000 RPM for 10 minutes at 4°C, removing the supernatant, then resuspending the cells in 100 μ L of water. In **Figure 3a**, 50 μ L of resuspended cells were added to individual wells in a clear 384-well plate. The cells were incubated in the plate for 24 hours to allow the cells to “develop” color on the bottom of the well. The plate was photographed upside down in normal light conditions. In **Figure 3b**, 5 μ L of resuspended cells were added to individual wells in a 1536-well plate and visualized upside down using a UV transilluminator without a filter. In **Figure 6**, Controls #1-8 were added at each evolutionary timepoint as a color reference for circuit function, but a lower concentration of cells (1.5 mL cells / 300 μ L of water) was used compared to the image shown in **Figure 3a**, resulting in a lighter color. For evolved populations, every 10 generations the deep 96-well plate was centrifuged at 2000 RPM for 10 minutes at 4°C, removing the supernatant, washing with 500 μ L of water, centrifuging again at 2000 RPM for 10 minutes at 4°C, removing the supernatant, then resuspending the cells in 100 μ L of water. 50 μ L of the resuspended cells were visualized as described above for **Figure 3a**.

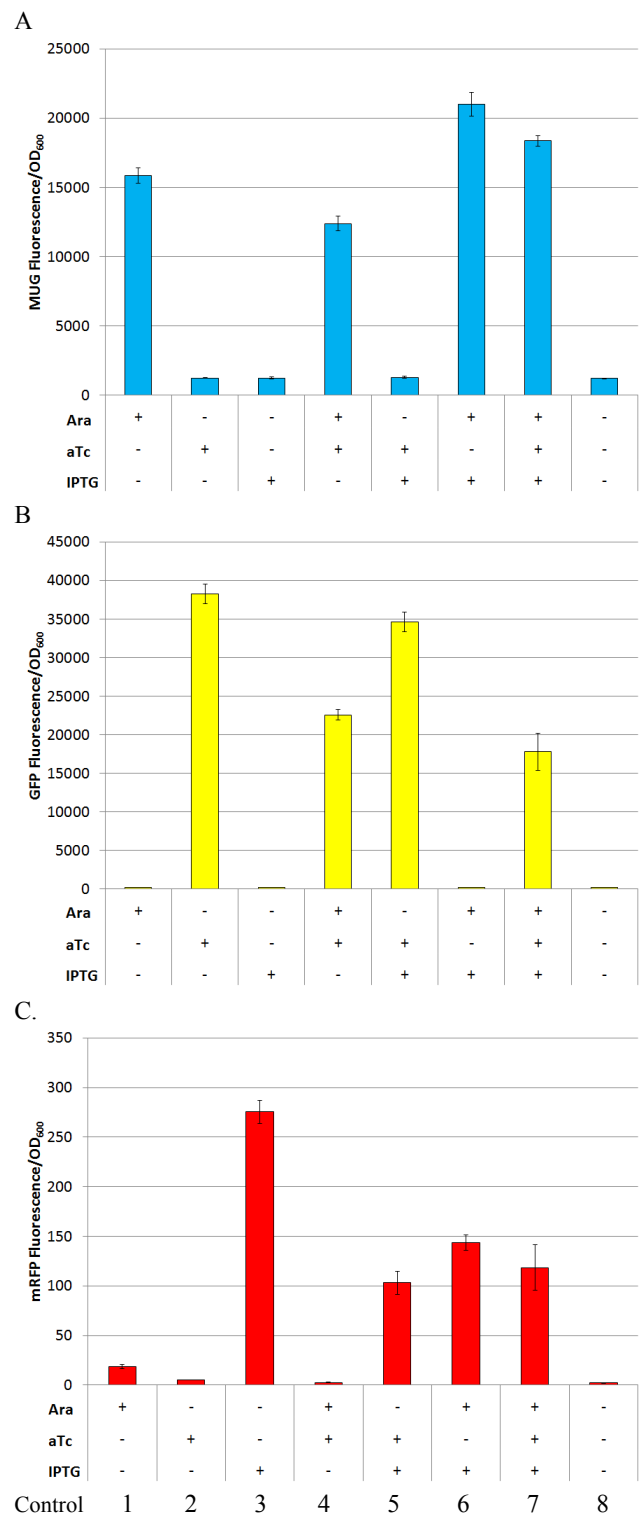


Figure 2. Independent and combinatorial expression of each color from the CMY circuit. Eight control cultures grown with different inducers and molecules (see Materials and Methods) were measured for (A) expression of LacZ, (B) expression of GFP, and (C) expression of mRFP. The inducers Ara, aTc, and IPTG express LacZ, GFP, and mRFP, respectively. Error bars represent one standard deviation from the mean of three independent replicates.

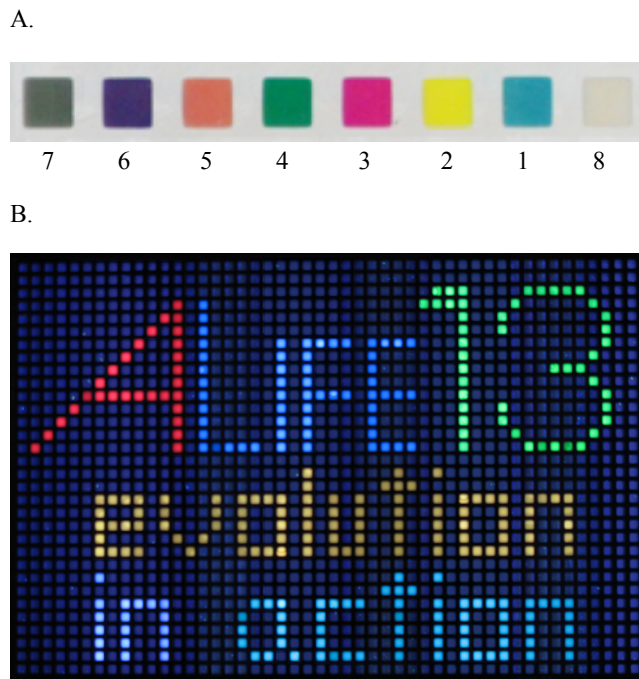


Figure 3. Visualization of CMY circuit colors under (A) normal light and (B) UV light. In normal light conditions, the color for controls #1-8 are visualized as described in Materials and Methods and the control numbers are labeled underneath each colored square well. In UV light, the color for controls #1-6 are visualized as described in Materials and Methods, as follows: “A” is written with control #3, “Life” with control #1, “13” with control #2, “evolution” with control #5, “in” with control #6, and “action” with control #4.

Results

Independent and combinatorial expression of each color from the CMY circuit. To test the functionality of the CMY circuit, we first performed a control experiment to measure expression of each color using different inducers and molecules (see Materials and Methods for details). The results of this experiment is shown in **Figure 2**. Starting with **Figure 2a**, the results show that LacZ is expressed about 10-fold above background levels with addition of arabinose, but not with aTc or IPTG. With other inducer combinations, LacZ is only expressed with arabinose, but to varying levels. This indicates that expression of other genes in the circuit affect expression of LacZ, possibly due to competition for expression and metabolic load. In **Figure 2b**, GFP is only expressed with the addition of aTc and is about 100-fold above background levels with other inducers. Like LacZ, GFP expression is also affected by expression of other genes in the circuit. **Figure 2c** shows that mRFP is expressed only with the addition of IPTG and is over 10-fold above background levels with other inducers. Overall, the results indicate the independent and combinatorial expression of each color in the CMY circuit, but expression levels are affected by different combinations of inducers. **Figure 3** shows visually that distinct colors are produced with each combination of inducer used, demonstrating combinatorial expression. Note that GFP appears yellow visually in **Figure 3a**, but is green in **Figure 3b** under UV light.

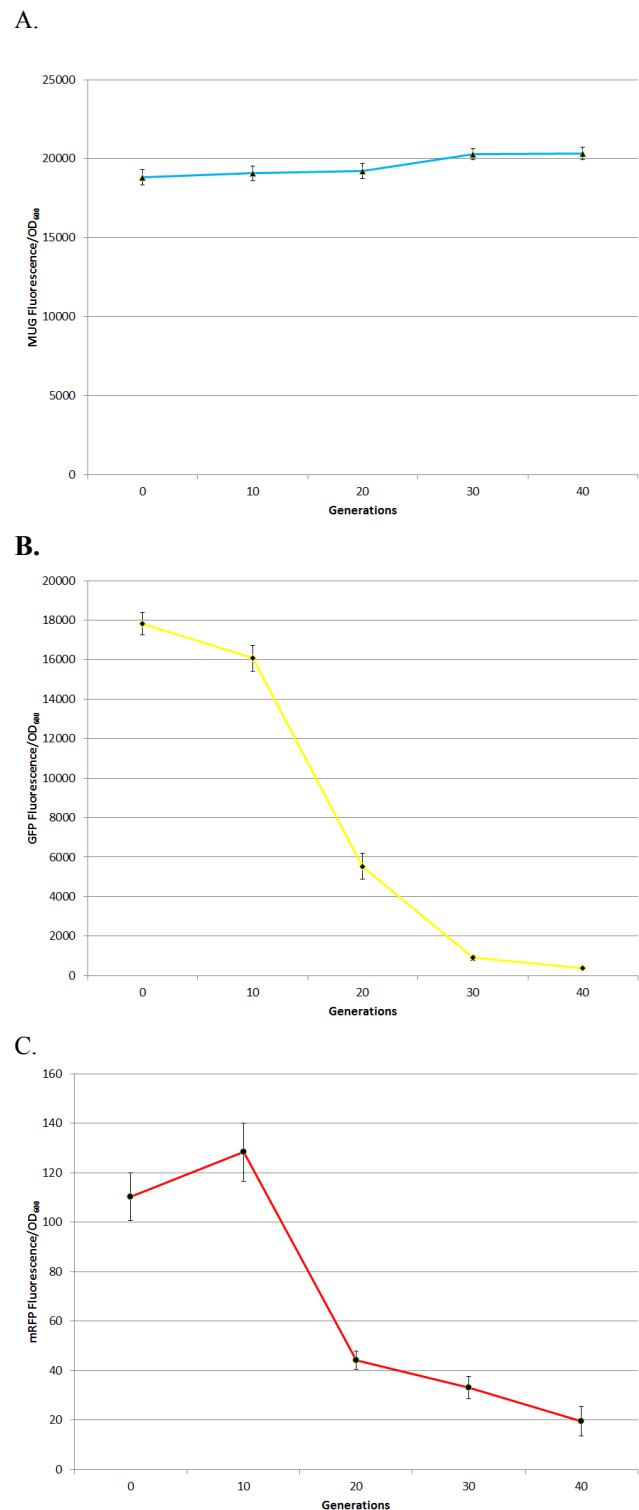


Figure 4. Evolutionary stability dynamics of the CMY circuit. Normalized fluorescence on the y-axis is plotted against the number of generations for (A) MUG fluorescence (LacZ), (B) GFP fluorescence, and (C) mRFP fluorescence. Error bars represent the standard deviation from the mean of 88 independent replicates.

Evolutionary stability dynamics of the CMY circuit.

Next, we performed an evolution experiment to measure the evolutionary stability dynamics of the CMY circuit and determine whether replicate populations evolved in parallel or divergently. For this experiment, we evolved 88 replicate populations in conditions where all colors in the circuit are turned on to increase the metabolic load and thereby maximize evolutionary processes. The results of this experiment are shown in **Figure 4**. **Figure 4a** shows the evolutionary stability dynamics of LacZ which remains relatively constant for 40 generations, with slightly increased expression at generation 40. In contrast, **Figures 4b and 4c** show that both GFP and mRFP expression is constant for about 10 generations with fluctuations, then both dramatically decrease by generation 20. Both GFP and mRFP expression continues to decrease slowly, but have not dropped to background levels by generation 40.

Loss-of-function mutation in the CMY circuit. To determine the mutation responsible for decreased GFP and mRFP expression, four evolved populations from the 40 generation timepoint were streaked out and a single clone from each was grown overnight for plasmid extraction. The mutant plasmid was shown to have a deletion between repeated B0015 terminators (129 bp) that effectively removed both the GFP and mRFP expression cassettes (**Figure 5**). The sequencing data was very clean, indicating that the plasmids within each clone sequenced were likely identical or nearly identical (e.g. noisy sequencing data may have indicated a mixture of different plasmids). This mutation was unsurprising since similar deletions between repeated terminators have occurred in other circuits we have studied (Sleight et al., 2010b). Although this result was somewhat expected, we still did not know exactly how a three-gene circuit would lose function. Interestingly, in a pilot evolution study, an earlier version of this circuit with different genetic parts that was transformed into a different strain, using different inducer concentrations than reported here, lost the cyan function first.

Visualizing color variation between 88 replicate evolved populations. We expected to see color differences between replicate populations over evolutionary time. In order to visualize the color of each evolved population at different evolutionary timepoints, we washed and resuspended the cells in water, then added these cells to individual wells in a 384-well plate. The results are shown in **Figure 6**. While there will be variation in color due to experimental methods (e.g. different number of cells put in each well) as well as variation due to the stochastic nature of evolutionary dynamics, the photos of each plate at different timepoints clearly show a color change from a dark greenish-brown to cyan color over the course of 40 generations. The sharpest transition in color is seen between generations 10 and 20, matching the quantitative data (**Figure 4**) closely. Generation 20 cells are a dark cyan color since there is still some GFP and mRFP being expressed. By generation 40, the cells appear to match the cyan color of the circuit when only LacZ is expressed. The cyan color would likely slowly fade away if the evolution experiment was continued beyond 40 generations.

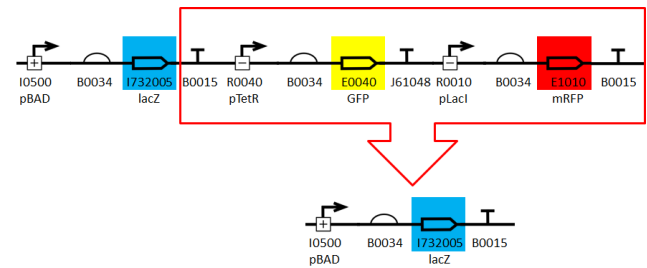


Figure 5. Dominant loss-of-function mutation in the CMY circuit. Individual clones from four independently evolved populations all have the same loss-of-function mutation: a deletion between repeated B0015 terminators that effectively removes the GFP and mRFP expression cassettes. The mutant clones can still express LacZ.

Discussion and Future Directions

In this study, we demonstrate the design and engineering of a functional CMY circuit, where each color in the circuit can be expressed independently and in combination. To measure the evolutionary stability dynamics of this prototype circuit, we evolved 88 independent populations with all colors turned on. We observed striking parallel evolution in the color change of the evolved populations that is in agreement with the quantitative measurements. The dominant mutation that we found in four of the populations is a deletion between repeated terminators that effectively removes the GFP and mRFP expression cassettes (note that there is no significant homology between GFP and mRFP). Although only clones in four of the 88 populations were sequenced, it is likely that the other populations had the same loss-of-mutation. Incidentally, a previous study found that a circuit with repeated B0015 terminators had a deletion between these sequences even when using a strain with a *recA* mutation and therefore replication slippage alone can cause this common mutation (Canton et al., 2008). Our previous work on the evolutionary stability of another genetic circuit with repeated B0015 terminators showed that nine out of nine populations lost function due to the same deletion between these repeated sequences, but re-engineering the circuit to have non-homologous terminators increases its evolutionary stability (Sleight et al., 2010b). The prototype CMY circuit in this study had repeated B0015 terminators only because this was the first version of the circuit tested to determine if the circuit was functional. We expect some future versions of this circuit to evolve more divergently (produce a wide variety of colors) due to the absence of repeated sequences. Also, it may be interesting to understand how the absence of different inducers in the media changes the evolutionary stability dynamics and loss-of-function mutations in future versions of the circuit. We have recently developed an assembly method to randomize parts (e.g. promoters, coding sequences, transcriptional terminators) to generate different combinations of genetic circuits. We will use this method to shuffle parts to engineer various CMY circuits, then use a directed evolution approach to evolve selected circuits individually and pooled to determine which versions of the circuit are most evolutionarily robust. We expect that some circuit variants will evolve in parallel and some will evolve divergently due to genotypic (e.g. repeat sequences, GC content, specific genetic elements) and phenotypic (e.g. metabolic load due to the

expression of foreign proteins individually and in A.



B.



C.



D.



E.

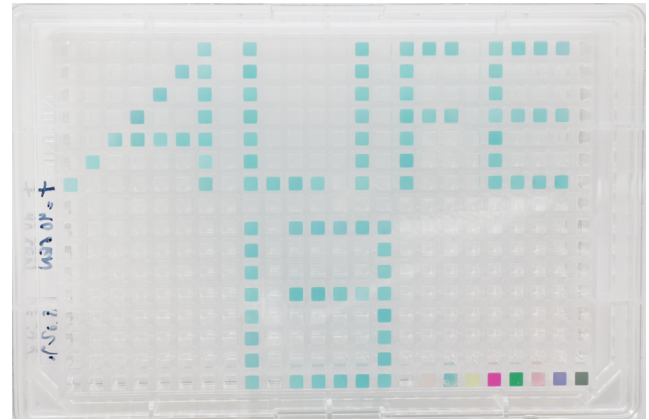


Figure 6. Color variation in evolved CMY circuit populations. Cells from each of the 88 independently evolved populations at different evolutionary timepoints were washed and resuspended in water, then added to individual wells in a 384-well plate to spell Alife13. Each plate represents (A) Generation 0, (B) Generation 10, (C) Generation 20, (D) Generation 30, and (E) Generation 40. The bottom right corner shows eight colored wells that represent the controls #1-8 (see Materials and Methods). The #8 control (unpigmented cells with no inducers) is shown on the left, then the #1-7 controls are shown going from left to right. Since the same cells were used for all controls on all plates, the color fades in the later timepoints.

combination) differences. The evolutionary stability dynamics and loss-of-function mutations will be measured in order to better understand parallel/divergent evolutionary processes and determine design principles for robust synthetic systems. Ideally a selective pressure will be used to maintain function of circuit components, but when that is not possible, the next best method is to lower expression level and mutation rate to a level that maintains function for as long as possible. If a high expression level is needed, then the next best method is to rationally design circuits that mutate in a predictable manner such that multiple versions of the circuit can turn on, then off via mutation. With this goal in mind, we also aim to identify a CMY circuit variant that will act as an “evolutionary timer circuit” for industrial applications when timed functions are needed, such that one color loses function after x generations, then a second color is lost after another y generations, and finally a third color loses function after z generations.

Acknowledgements

We would like to thank BEACON: An NSF Center for the Study of Evolution in Action for funding this research. Also, we would like to thank members of the Sauro Lab (especially Wilbert Copeland and Bryan Bartley) and Klavins Lab (especially Rob Egbert) for use of equipment, useful materials, and discussions.

References

- Anderson, J Christopher, Christopher a Voigt, and Adam P Arkin. 2007. "Environmental signal integration by a modular AND gate." *Molecular systems biology* 3 (133) (January): 133. doi:10.1038/msb4100173.
- Balagaddé, Frederick K, Lingchong You, Carl L Hansen, Frances H Arnold, and Stephen R Quake. 2005. "Long-term monitoring of bacteria undergoing programmed population control in a microchemostat." *Science (New York, N.Y.)* 309 (5731) (July): 137-40. doi:10.1126/science.1109173.
- Basu, S, Y Gerchman, C H Collins, F H Arnold, and R Weiss. 2005. "A synthetic multicellular system for programmed pattern formation." *Nature* 434 (7037): 1130-1134. doi:nature03461 [pii] 10.1038/nature03461.
- Canton, Barry, Anna Labno, and D Endy. 2008. "Refinement and standardization of synthetic biological parts and devices." *Nat Biotechnol* 26 (6).
- Colosimo, Pamela F, Kim E Hosemann, Sarita Balabhadra, Guadalupe Villarreal, Mark Dickson, Jane Grimwood, Jeremy Schmutz, Richard M Myers, Dolph Schluter, and David M Kingsley. 2005. "Widespread parallel evolution in sticklebacks by repeated fixation of Ectodysplasin alleles." *Science* 307 (5717): 1928-33. doi:10.1126/science.1107239.
- Cooper, V S, D Schneider, M Blot, and R E Lenski. 2001. "Mechanisms causing rapid and parallel losses of ribose catabolism in evolving populations of Escherichia coli B." *J Bacteriol* 183 (9): 2834-2841. doi:10.1128/JB.183.9.2834-2841.2001.
- Elena, S F, and R E Lenski. 2003. "Evolution experiments with microorganisms: the dynamics and genetic bases of adaptation." *Nat Rev Genet* 4 (6): 457-469.
- Elowitz, M B, and S Leibler. 2000. "A synthetic oscillatory network of transcriptional regulators." *Nature* 403 (6767) (January): 335-8. doi:10.1038/35002125.
- Entus, R, B Aufderheide, and H M Sauro. 2007. "Design and implementation of three incoherent feed-forward motif based biological concentration sensors." *Systems and Synthetic Biology* 1 (3): 119-128.
- Gardner, T S, C R Cantor, and J J Collins. 2000. "Construction of a genetic toggle switch in Escherichia coli." *Nature* 403 (6767): 339-342. doi:10.1038/35002131.
- Gibson, D G, J I Glass, C Lartigue, V N Noskov, R Y Chuang, M A Algire, G A Benders, et al. 2010. "Creation of a Bacterial Cell Controlled by a Chemically Synthesized Genome." *Science*. doi:science.1190719 [pii] 10.1126/science.1190719.
- Herring, C D, A Raghunathan, C Honisch, T Patel, M K Applebee, A R Joyce, T J Albert, et al. 2006. "Comparative genome sequencing of Escherichia coli allows observation of bacterial evolution on a laboratory timescale." *Nat Genet* 38 (12): 1406-1412.
- Knight, T. 2003. "Idempotent Vector Design for Standard Assembly of Biobricks." *DSpace*
- Levskaya, A, A A Chevalier, J J Tabor, Z B Simpson, L A Lavery, M Levy, E A Davidson, et al. 2005. "Synthetic biology: engineering Escherichia coli to see light." *Nature* 438 (7067): 441-442. doi:nature04405 [pii] 10.1038/nature04405.
- Meyer, Justin R, Anurag A Agrawal, Ryan T Quick, Devin T Dobias, Dominique Schneider, and Richard E Lenski. 2010. "Parallel changes in host resistance to viral infection during 45,000 generations of relaxed selection." *Evolution; international journal of organic evolution* 64 (10) (October): 3024-34. doi:10.1111/j.1558-5646.2010.01049.x.
- Philip, Daniel S., Derek S. Sarovich, and John M. Pemberton. 2009. "Complete sequence and analysis of the stability functions of pPSX, a vector that allows stable cloning and expression of Streptomycete genes in Escherichia coli K12." *Plasmid* 62 (1) (July): 39-43. doi:10.1016/j.plasmid.2009.03.002.
- Riehle, M M, A F Bennett, and A D Long. 2005. "Changes in gene expression following high-temperature adaptation in experimentally evolved populations of E. coli." *Physiol Biochem Zool* 78 (3): 299-315.
- Schoustra, S E, A J Debets, M Slakhorst, and R F Hoekstra. 2006. "Reducing the cost of resistance; experimental evolution in the filamentous fungus Aspergillus nidulans." *J Evol Biol* 19 (4): 1115-1127.
- Shetty, R P, D Endy, and T F Knight Jr. 2008. "Engineering BioBrick vectors from BioBrick parts." *J Biol Eng* 2: 5.
- Sleight, Sean C, Bryan A Bartley, Jane A Lieviant, and Herbert M Sauro. 2010a. "In-Fusion BioBrick assembly and re-engineering." *Nucleic Acids Research* 38 (8): 2624-2636.
- . 2010b. "Designing and engineering evolutionary robust genetic circuits." *Journal of biological engineering* 4 (1) (November): 12. doi:10.1186/1754-1611-4-12.
- Sleight, Sean C, Christian Orlic, Dominique Schneider, and Richard E Lenski. 2008. "Genetic basis of evolutionary adaptation by Escherichia coli to stressful cycles of freezing, thawing and growth." *Genetics* 180 (1) (September): 431-43. doi:10.1534/genetics.108.091330.
- Sleight, Sean C, and Richard E Lenski. 2007. "Evolutionary adaptation to freeze-thaw-growth cycles in Escherichia coli." *Physiological and biochemical zoology PBZ* 80 (4): 370-385.
- Toprak, Erdal, Adrian Veres, Jean-Baptiste Michel, Remy Chait, Daniel L Hartl, and Roy Kishony. 2012. "Evolutionary paths to antibiotic resistance under dynamically sustained drug selection." *Nature genetics* 44 (1) (January 18): 101-5. doi:10.1038/ng.1034.
- Tyo, K E, P K Ajikumar, and G Stephanopoulos. 2009. "Stabilized gene duplication enables long-term selection-free heterologous pathway expression." *Nat Biotechnol* 27 (8): 760-765. doi:nbt.1555 [pii] 10.1038/nbt.1555.

Design and construction of a prototype CMY (Cyan-Magenta-Yellow) genetic circuit as a mutational readout device to measure evolutionary stability dynamics and determine design principles for robust synthetic systems

- Vidal-Aroca, Faustino, Michele Giannattasio, Elisa Brunelli, Alessandro Vezzoli, Paolo Plevani, Marco Muzi-Falconi, and Giovanni Bertoni. 2006. "One-step high-throughput assay for quantitative detection of β -galactosidase activity in intact Gram-negative bacteria, yeast, and mammalian cells." *BioTechniques* 40 (4) (April): 433-440. doi: 10.2144/000112145.
- Yoon, Sang-Hwal, Ju-Eun Kim, Sook-Hee Lee, Hye-Min Park, Myung-Suk Choi, Jae-Yean Kim, Si-Hyoung Lee, Yong-Chul Shin, Jay D Keasling, and Seon-Won Kim. 2007. "Engineering the lycopene synthetic pathway in *E. coli* by comparison of the carotenoid genes of *Pantoea agglomerans* and *Pantoea ananatis*." *Applied microbiology and biotechnology* 74 (1) (February): 131-9. doi:10.1007/s00253-006-0623-z.
- You, L, R S Cox 3rd, R Weiss, and F H Arnold. 2004. "Programmed population control by cell-cell communication and regulated killing." *Nature* 428 (6985): 868-871.

The Humanities and ALife

Papers

Using Pictures to Visualize the Complexity of Gene Regulatory Networks

Sylvain Cussat-Blanc and Jordan Pollack

DEMO Lab, Volen National Center for Complex System - Brandeis University MS018
415 South street, Waltham, MA 02454, USA
cussat@brandeis.edu, pollack@brandeis.edu

Abstract

This paper proposes a new method to evaluate the complexity of a Gene Regulatory Network (GRN). It is based on the generation of pictures. In addition to being visually interesting, the pictures show the capacity of the GRN to produce smooth and/or sudden transitions, fractal-like complexity and regularities. We also have studied the influence of the size of the GRN on the complexity of pictures generated.

Introduction

In nature, the development processes are able to produce very large and very complex structures. Based on cells driven by a gene regulatory network, the growth process is able to produce organisms composed of billions of specialized cells organized so that they can act in their environment. Over the past years, many researchers in the field of artificial embryogenesis have proposed various developmental models more or less biologically plausible. These works are mainly based on gene regulation with two leading models (Eggenberger, 1997; Banzhaf, 2003). However, if we only focus on the generation of morphologies (or shapes) with specialization, the results are limited in comparison to what nature is able to produce. One of the best results consists in developing a 2-D or 3-D colored shapes, where the colors represent the cell specialization (Joachimczak and Wróbel, 2008; Doursat, 2008; Cussat-Blanc et al., 2011).

Our main project is to use a cell-based developmental model to generate robot morphologies. A cellular model is used to develop an artificial organism evaluated in a physics simulator (Cussat-Blanc and Pollack, 2012). A gene regulatory network controls the behavior of the cells. It allows the cells to orient their division plan, to differentiate to a particular cell type or to choose between a symmetric division (no cell specialization) or an asymmetric one (one cell is specialized whereas the second one is unspecialized). With this approach, we already were able to generate interesting robot morphologies, as presented in figure 1, that are currently under-construction with real robotic units.

In our opinion, a GRN is well suited for this range of problems because it is biologically plausible. Because nature



Figure 1: Examples of robot morphologies generated by the use of a cell-based developmental controlled by a gene regulatory network.

proves that this approach works, we can expect them to scale up better than other existing methods. However, for now, the morphologies are far from what nature is able to produce. To try to understand why an artificial Gene Regulatory Networks (GRN) cannot produce shapes as complex as a real regulatory network, we propose in this paper to focus on the regulatory network itself and to remove the cell-based developmental model usually plugged to this system. Instead, the genotype-phenotype mapping translates pixel addresses to colors. We call it a pixel mapping. The earliest use we know of involved imaging is the results of learning on the Intertwined Spiral problem (Fahlman, 1990).

Many generative methods exist to generate pictures. They took inspiration from Karl Sims' work in which he used a blind watchmaker to evolve symbolic expression rules to produce images (Sims, 1991). The closest approach to our must be the Secretan et al.'s CPPN-based approach (Secretan et al., 2008). They propose an online tool to generate pictures. In a CPPN, the coordinates of a unit (here a pixel) are used to modify the weights of a NEAT network. For picture generation, the output of the neural network evolved by the NEAT algorithm is the pixel color. With same objective, David Hart used genetic programming to generate interesting pictures (Hart, 2007). His approach is based on a set of predefined functions that an evolutionary algorithm combines. Once again, the coordinates of the pixels are used as inputs of the systems. Romero and Machado propose a full state-of-the-art of evolutionary art in (Romero and Machado, 2007). In this review, many other approaches are presented.

In this work, we have used a GRN to generate pictures. The results we have obtained were unexpected: the pictures generated are very complex, with or without regularities and are surprisingly aesthetic. The GRN can generate various complex structures in the same picture, producing smooth or sudden transitions between the colors. Some fractal-like properties have also been observed in many pictures.

This paper is organized as follow. The next section introduces the functioning of a real gene regulatory network. It also details our implementation of the regulatory network. Then, we propose a method to use the regulatory network to generate pictures. We also present the blind watchmaker approach used to evolve our regulatory network. Next, we present a set of pictures obtained with our system. The discussion describes the capacity of the GRN and proposes a study of the influence of the size of the GRN on the complexity of the pictures. Finally, the paper concludes on the future work opened by this approach.

Gene Regulatory Network

Background on artificial regulatory networks

Many current developmental models rely on an artificial GRN's to simulate cell differentiation. These systems are more or less inspired by gene regulation systems of living systems. In living systems, the cells of an organism have several functions. They are described in the organism genome and their expressions are controlled by a regulatory network (Davidson, 2006). Cells use external signals collected from protein sensors localized on the membrane to activate or inhibit the transcription of the genes. The gene expressions determine the cells' behaviors.

Eggenberger first used a GRN to generate a 3-D organism able to move in its environment by modifying its morphology (Eggenberger, 1997). Reil then proposed a model biologically plausible with a genome defined as a vector of numbers (Reil, 1999). Here, each gene starts with the sequence (0101), named the "promoter". Then, a graph is used to visualize the gene activations and inhibitions over time with networks randomly generated. Observations revealed the existence of various patterns such as gene activation sequencing, chaotic expressions or cyclic expressions. The author also pointed out that the system was resistant to randomly deteriorations of the genomes. Banzhaf also described an artificial GRN model close to real-world gene regulation (Banzhaf, 2003), detailed further bellow. Starting from these seminal models, many variations have been explored in order to address various concerns and applications. Several works addressed artificial embryogeny problems with models of GRN ranging from cellular automaton modeling (Chavoya and Duthen, 2008) to stripped-down version of GRN combined with complex developmental systems (Joachimczak and Wróbel, 2008; Doursat, 2008). Some works have also addressed control problems: using GRN as a control function to map a virtual robot's

sensory inputs to its motor actuator values. This has been applied in various setup, from foraging agents (Joachimczak and Wróbel, 2010) to pole balancing (Nicolau et al., 2010).

Our implementation of the regulatory network

We have based our regulatory network on Banzhaf's model (Banzhaf, 2003). He designed it to be as close as possible to a real gene regulatory network. As DNA is composed of a sequence of nucleotides, Banzhaf's network is encoded within a sequence of bits. As a real gene starts with the particular sequence of nucleotides e.g. *TATA*, a gene in Banzhaf's network starts with a particular sequence of 8 bits named the "promoter". A gene is then encoded next to this sequence by five 32-bit integers, named the "sites". This mechanism allows the generation of a variable number of genes in a fixed size chromosome. However, as in nature, it also generates a certain amount of noncoding DNA, the probability to have a promoter being very low (2^{-8}). This noncoding DNA¹ is thought to be used in nature to protect the genome from mutation by lowering the probability that a mutation will affect a coding nucleotide.

Banzhaf's model has been neither designed to be evolved nor to control any kind of agent. However, Nicolau used an evolution strategy to evolve the GRN to control a pole-balancing cart (Nicolau et al., 2010). Even if the cart has shown consistent behaviors, the evolution of the GRN has been an issue. In our opinion, the difficulty of the evolution is due to: (1) the *noncoding DNA* and (2) the *dynamics* of the network. According to these observations, we have decided to modify the encoding of the regulatory network and its dynamics. In our model, a gene regulatory network is defined as a set of proteins. Each protein has the following properties:

- The protein *identifier* coded as an integer between 0 and p . The upper value p of the domain can be changed in order to control the precision of the GRN. In Banzhaf's work, p is equivalent to the size of a site, which is 32 bits. We have kept the same precision by setting up p to 32.
- The *enhancer identifier* coded as an integer between 0 and p . The enhancer identifier is used to calculate the enhancing matching factor between two proteins.
- The *inhibiter identifier* coded as an integer between 0 and p . The inhibitor identifier is used to calculate the inhibiting matching factor between two proteins.
- The *type* determines if the protein is an *input* protein (which concentration is given by the environment of the GRN and which regulates other proteins but is not regulated), an *output* protein (which concentration is used as output of the network and which is regulated but does not regulate other proteins) or a *regulatory* protein (internal protein that regulates and is regulated by other proteins).

¹98% of human DNA

This encoding removes the problem of noncoding DNA of Banzhaf's approach. Each integer is used in the regulatory network and a modification of one of them will automatically imply a modification of the network.

The dynamics of the GRN is calculated as follow. First, the affinity of a protein a with another protein b is given by the enhancing factor u_{ab}^+ and the inhibiting u_{ab}^- :

$$u_{ab}^+ = p - |enh_a - id_b| ; \quad u_{ab}^- = p - |inh_a - id_b|$$

where id_x is the identifier, enh_x is the enhancer identifier of protein x and inh_x is the inhibiting identifier.

The GRN's dynamics is calculated by comparing the proteins two by two using the enhancing and the inhibiting matching factors. For each protein of the network, the global enhancing value is given by the following equation:

$$g_i = \frac{1}{N} \sum_j c_j e^{\beta u_{ij}^+ - u_{max}^+} ; \quad h_i = \frac{1}{N} \sum_j c_j e^{\beta u_{ij}^- - u_{max}^-}$$

where g_i (resp. h_i) is the enhancing (resp. inhibiting) value for a protein i , N is the number of proteins in the network, c_j is the concentration of protein j and u_{max}^+ (resp. u_{max}^-) is the maximum enhancing (resp. inhibiting) matching factor observed. β is a control parameter described hereafter.

The final modification of protein i concentration is given by the following differential equation:

$$\frac{dc_i}{dt} = \frac{\delta(g_i - h_i)}{\Phi}$$

where Φ is a function that keeps of the sum of all protein concentrations equal to 1.

β and δ are two constants that set up the speed of reaction of the regulatory network. The higher these values, the more sudden the transitions in the GRN. The lower they are, the smoother the transitions are.

Whereas the input proteins of a GRN can be used to describe the current state of the environment, the output proteins select the level of application of each possible action. The network can also be easily encoded in a genome to be evolved by an evolutionary algorithm. The next section presents how the GRN is used to generate pictures and how it is encoded in a genome.

Picture generation

Binding between a GRN and a picture

To generate a picture with a GRN, the GRN calculates the RGB color of each pixel of the picture. To do so, the GRN has two inputs that correspond to the coordinates of the current pixel and three outputs, one for each color component. The coordinate (x, y) of a pixel are transformed into proteins concentrations so that they do not overflow the network:

$$c_x = \frac{0.1x}{width} ; \quad c_y = \frac{0.1y}{height}$$

where c_x (resp. c_y) is the concentration of the protein associated to the abscissa x (resp. the ordinate y) of the current pixel, $width$ and $height$ define the size of the picture.

The resulting RGB component values are given by the following equations:

$$out_r = \frac{255 * c_r}{max_r} ; \quad out_g = \frac{255 * c_g}{max_g} ; \quad out_b = \frac{255 * c_b}{max_b}$$

where out_r (resp. out_g and out_b) is the value of the red (resp. green and blue) component for the current pixel, c_r (resp. c_g and c_b) is the concentration of the output protein associated to the red (resp. green and blue) component in the GRN (this concentration is always between 0 and 1) and max_r (resp. max_g and max_b) is the maximum concentration observed in the picture for the red (resp. green and blue) component.

Before the generation of the picture, the GRN is first evolved for 100 steps without any inputs in order to stabilize the concentration. This is a very common technique because the GRN are known to oscillate during the first steps. After this initialization, the GRN is duplicated for each pixel of the picture and the duplicated GRN's are run for 25 more steps with the inputs corresponding to their pixels. The pixel colors are then calculated as explained before.

Encoding of the GRN

To be evolved by an evolutionary algorithm, the GRN is encoded into a genome with two independent chromosomes. The first chromosome encodes the set of proteins and the second one encodes the parameters of the dynamics β and δ .

Because a GRN can have a variable number of proteins, the first chromosome is defined as a variable length chromosome of indivisible proteins. Each protein is encoded within four integers: three between 0 and p for the three different identifiers and one in $[0, 2]$ for the type of the protein.

If an evolutionary algorithm has to evolve this chromosome, the modification operators have to be redefined. First, the *crossover* consists in exchanging subparts of two different networks. Because proteins are indivisible, the crossover points have to be chosen between two proteins. It ensures the integrity of each sub-network. The local connectivity is thus kept. Only new links between the different sub-networks are created. The *mutation* can be applied in three equiprobable ways: *mutating an existing protein* by randomly changing one of its four integers, *adding a new protein* randomly generated or *removing one random protein* from the network.

In this work, the chromosome is ordered as following: (1) the first two proteins are two inputs proteins that correspond to the coordinate of the pixel, (2) the three next proteins are the three output proteins: one for the red component, one for the green and one for the blue, (3) the remaining proteins are only regulatory proteins. Because one of the objective if the study of the impact of the size of the regulatory network on the complexity of its behavior, the size of this chromosome

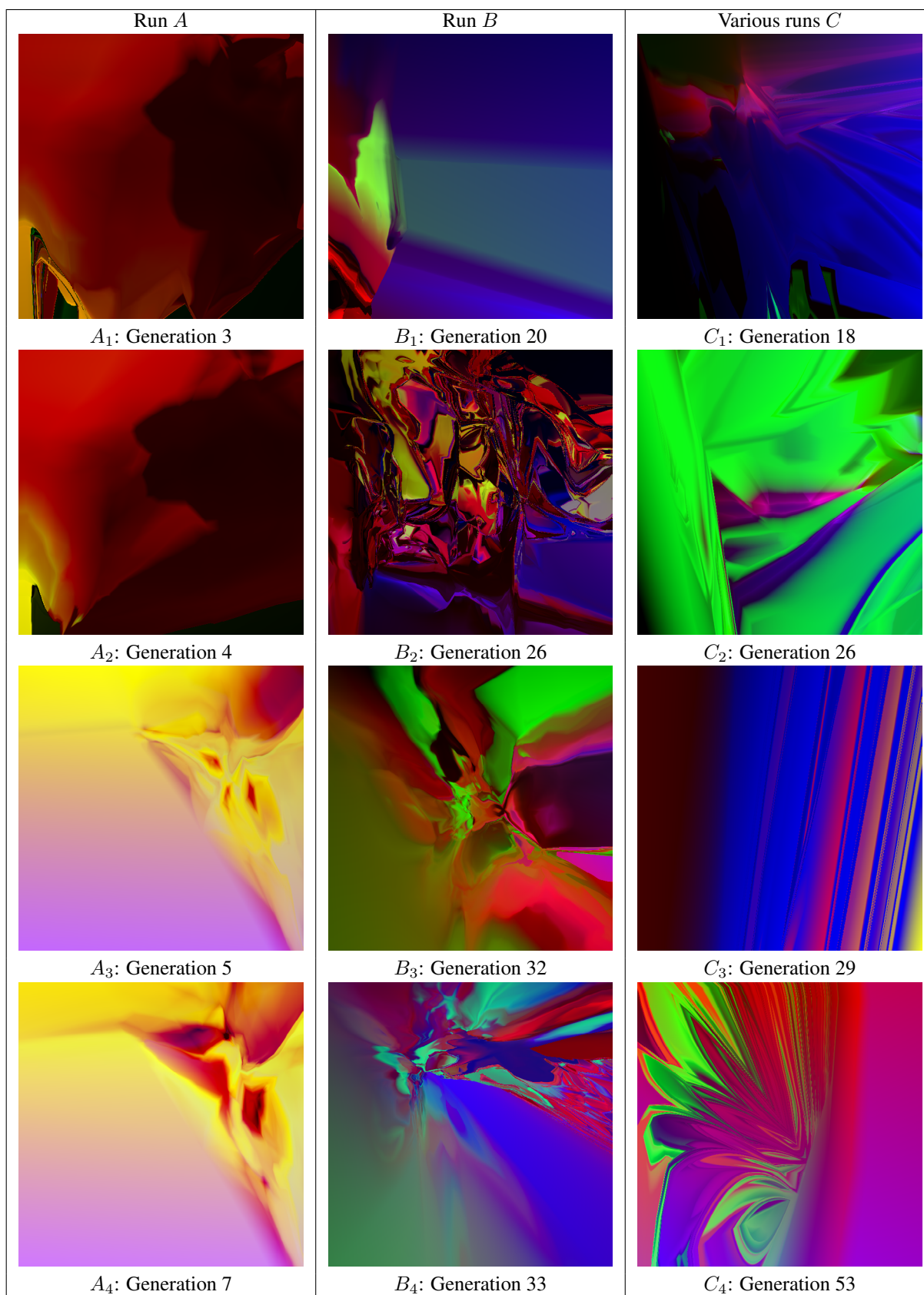


Figure 2: Examples of generated pictures with 12 regulatory proteins in the GRN taken in the same run (first 2 columns) or in various runs (last column)

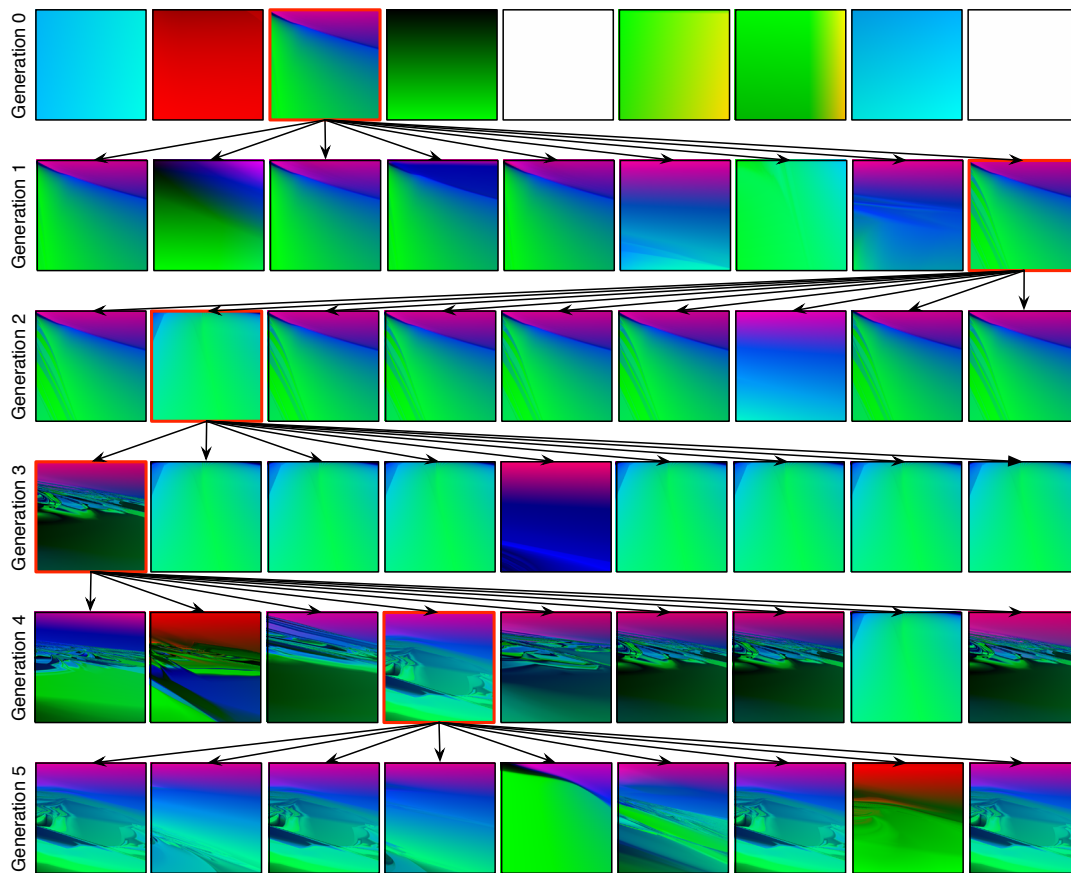


Figure 3: Example of the first five generations of a run with the blind watchmaker. It shows the fast compexification of the behaviors generated by the regulatory networks.

has been fixed and only the mutation of existing proteins is applied. All the experimentations presented hereafter give the corresponding numbers of proteins.

The second chromosome only contains the constants β and δ . It is defined by a chromosomes that contains 2 float values. These values can evolve between 0.5 and 2. These bounds have been empirically chosen. If the values are less than 0.5, the GRN stays stationary. With high values, the GRN behavior is usually chaotic.

To evolve the GRN, we use a "Blind Watchmaker" interactive evolutionary algorithm, described in the next section.

Interactive evolution of the pictures

The blind watchmaker is a common name given to an interactive evolutionary method first proposed in 1986 by Richard Dawkins (Dawkins, 1986). He originally used this method to sustain the theory of natural evolution using a pedagogical model called *biomorphs*, fractal-like creatures generated with a small set of genes. This method gave birth more recently to the field of interactive evolution. Many applications are nowadays based on this principle to solve various problems. For example, it has been used with genetic

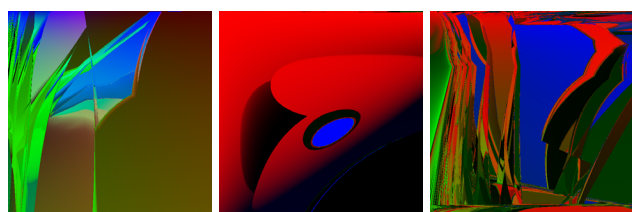
programming to generate realistic camouflage (Reynolds, 2011), or with HyperNEAT to generate 2-D pictures (Secrean et al., 2008) or 3-D shapes (Clune et al., 2010).

In this work, we first generate 9 random genomes. The 9 corresponding pictures are then produced and proposed to the user. The user can then save the GRN's that have generated pictures he likes and select one of the 9 pictures to be evolved. When a GRN is selected, the application generates 9 new pictures by mutating 10% of the selected GRN's genome. We have decided not to use the crossover operator to enhance the diversity of generated pictures. For the same reason, the mutation rate has been deliberately chosen high. With this method, we have generated a pool of diversified pictures. Next section presents some of them and discusses the properties of the GRN, which generate these pictures.

Results and discussions

Study of the complexity of the GRN

In order to visualize the complexity of the outputs generated by the GRN, we first used a GRN composed of 12 regulatory proteins (in addition to the 2 inputs proteins and the 3 output ones). With the blind watchmaker, we have evolved a set of



(a) Generation 16 (b) Generation 19 (c) Generation 23

Figure 4: Examples of pictures generated with GRN's that contain 6 regulatory proteins

random regulatory networks. Figure 2 shows some pictures obtained with this approach. These pictures have been selected in two runs of the blind watchmaker in the first two columns (one run by column) and in various runs in the last columns.

First, we can observe the variety of the pictures obtained, as well with different seeds (columns) or during one seed's evolution (rows). Figure 3 shows the smooth changes generations after generations, even with a high mutation rate.

The complexity can also be visualized by the capacity of the GRN to produce smooth transitions between the colors such as on the pictures A_3 and B_3 of figure 2 or very sudden changes such as on picture B_2 . Many pictures also present both type of transition such as A_1 , B_4 or C_4 . It shows the capacity of the GRN to produce very different kinds of behaviors even with smooth modification of the inputs, a shift of one pixel in a direction producing a very small modification of one input protein.

The GRN is also able to produce this complexity in very few generations (usually, about 15 to 20 generations are necessary to obtain very complex pictures). Once the first complex picture is obtained, the complexity does not increase, visually speaking. The high mutation rate allows a large diversity of generated pictures, even if the GRN seems to be converged: in few generations, the blind watchmaker is able to generate new pictures completely different from the previous generations.

Finally, some pictures present regularities, such as picture C_3 or C_4 of figure 2. The same patterns are repeated

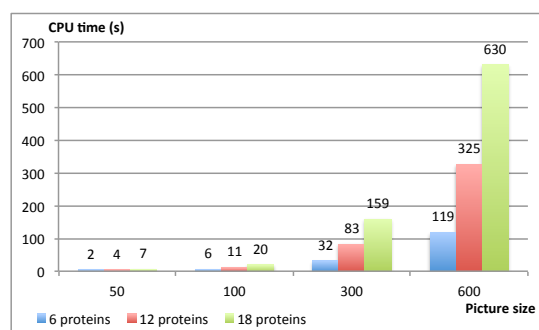


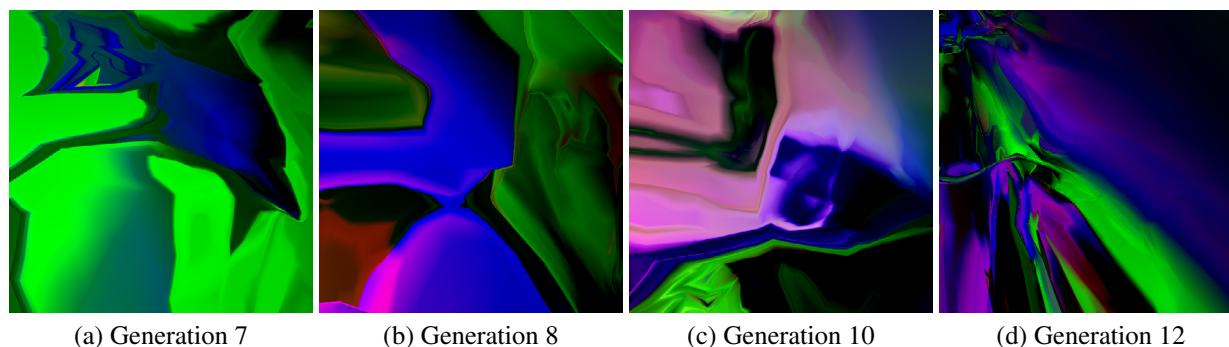
Figure 6: CPU time needed to generate an image with a GRN in function of the size of the image and the number of regulatory proteins in the GRN.

many times with few variations. For example, in picture C_3 , the same strips are repeated with a variation of width but with close colors. In picture C_4 , ovoid leaf-like shape are repeated with a rotation around a central point. This property is very important because it can explain the capacity of a GRN to produce repeated sequence of action with small variations. It shows how a GRN can produce in living organisms multiple legs, branches or any kinds of complex organ.

Influence of the size of the GRN on the complexity

In the previous experimentation, the number of the regulatory proteins has been arbitrarily chosen equal to 12. This value has been determined so that the pictures generated are interesting enough while keeping the GRN's size reasonable to maintain the interactivity with the user. To understand the importance of the size of the GRN, we have decided to generate pictures with GRN that have 6 and 18 regulatory proteins. The more complex pictures obtained are presented in figure 4 for GRN's with 6 regulatory proteins and figure 5 for GRN's with 18 regulatory proteins. Here, the pictures are taken from different runs.

The complexity of the pictures obtained is comparable with the different tested sizes of GRN's. However, with only 6 regulatory proteins, it was harder to generate images with



(a) Generation 7 (b) Generation 8 (c) Generation 10 (d) Generation 12

Figure 5: Examples of pictures generated with GRN's that contain 18 regulatory proteins

smooth color transitions. As presented in figure 4, the pictures have more sudden color transitions, which can be a limitation when the GRN's are used as behavior generators. With 18 regulatory proteins, the same kind of pictures is generated as with 12 regulatory proteins. However, a bigger GRN seem to generate complexity faster than a smaller one: in all the runs we have made with 18 regulatory proteins, 5 to 10 generations were necessary to obtain interesting pictures instead of 15 to 20 with 12 regulatory proteins.

The increase of the size of the GRN seems to reduce the time necessary to obtain complex behaviors. However, the computation time is also impacted by an increase of the size of the GRN. As presented in figure 6, the CPU time increases as well with the size of the pictures as with the number of regulatory proteins. In this experimentation, we have used a 3.16GHz Intel Xeon CPU. The values presented here represent an average of 50 runs made on randomly chosen GRN's obtained during different interactive evolution runs.

The main issue with the increase of the computation duration is the loss of interactivity of the software. It is important to find a good balance between the size of the pictures presented in the blind watchmaker and the number of regulatory proteins. In our experience, a GRN that contains 12 regulatory proteins is sufficient to generate interesting pictures. A GRN with 18 regulatory proteins generates the same kind of pictures but in fewer generations. Concerning the size of the picture, a 50x50 picture makes the appreciation the picture difficult but is sufficient to appreciate its complexity. A 100x100 is already sufficient to observe some details.

Scalability of the approach

An interesting property of this approach is that the images are scalable: if a user likes a picture, it can be easily enlarged by running the same GRN at a higher resolution. The same picture will be generated with more details. This property can be illustrated by figure 7 where we have zoomed on a specific region of a picture generated by evolution. We have zoomed in three steps 216 times from the original picture (on the left) to the last picture (on the right).

Zooming allows more and more details on the picture to appear. Transitory states of the regulatory network seem to be very complex. Some of them seem to have fractal property, such as the top purple-yellow transition on the right side picture. Even zoomed 216 times, a lot of details are invisible on the transition, some red pixels appearing at different points of the transition.

This quantity of details has to be compared with the size of the GRN's encoding. Indeed, each protein is encoded with 4 integers (3 for the identifiers and 1 for the protein type). Because these integers are between 0 and 32, 4 short integers are sufficient to encode a protein. Thus, it can be encoded with 4 bytes. The size of a GRN is then $4 * nbProt + 16$ bytes. The 16 bytes added correspond to the two double floating-point values that encode to the constants β and δ used to control the GRN's dynamics. In this experimentation, the GRN contains 17 proteins (2 inputs, 3 outputs and 12 regulatory proteins). Thus, the size of the GRN is 84 bytes, which is extremely low in comparison to all existing picture formats and the details generated by the GRN's. The GRN could be evolved to generate a given picture. It would produce a powerful compression algorithm, related to the IFS fractals of Barnsley (Barnsley, 1988).

Conclusion and perspectives

In this paper, we have used a gene regulatory network to generate pictures. We have used a direct encoding between the GRN and the pictures. The GRN provides the RGB values of each pixel of the picture according to its coordinates. This direct encoding is very common in literature (Sims, 1991; Hart, 2007; Secretan et al., 2008). The interesting results about using is a GRN instead of a CPPN or genetic programming is that the complexity of the generated pictures is inherent to the GRN. No function is used to control the input of the network. Moreover, the GRN's were able to produce fractal pattern and regularities in many pictures, which can be an interesting property when used to generate robot plans.

While there are other candidates for generative represen-

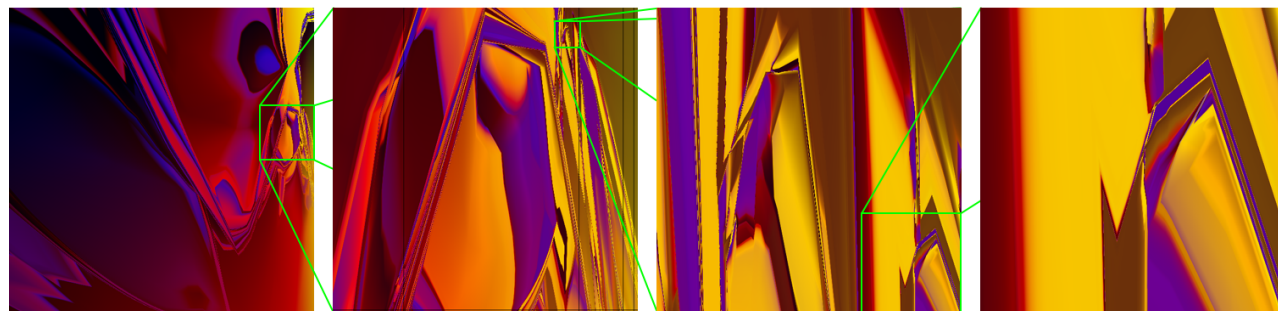


Figure 7: Example of the scalability of generated pictures. The picture on the left side is the original one, evolve with 12 regulatory proteins in 28 generations. The second picture is an enlargement of the first one. It is extended 6 times. The third picture is zoom 12 times on the second picture and the last one is zoomed 3 times on penultimate one.

tations, such as grammars, L-systems or HyperNEAT, we believe that GRN's are the most authentic representation coming from nature. Due to their high non-linearity, they are impossible to design and must be evolved. We have shown that evolution can be effective in a blind watchmaker setting, and that artificial GRN's have utility both in generating robotic body plans as well as interesting images.

The ease with which complex behaviors are obtained is surprising. Whereas most of existing approaches need many generations to obtain them, few are necessary with the GRN. The excessive complexity generated by nonlinear dynamical systems like GRN's is both a blessing and a curse. It enables the evolution of highly complex and multifaceted structures in nature, but gaining control over the process computationally has proven to be fraught with difficulty.

If we want the system to be really usable for an artistic purpose, the generation time of the pictures has to be improved. Currently, only 50x50 thumbnails are generated to keep the evolution interactive. With a GRN that contains 12 regulatory proteins, it takes about 45 seconds to generate the 9 pictures. Even if the application is multithreaded so that it divides the generation time by the number of cores, it is still the main limitation of the approach. However, the regulatory network could be easily transformed into matrix computing and, then, deployed on a graphics card. In this case, the computational time would be strongly reduced.

In conclusion, as a field, Artificial Life should reflect, algorithmically, on the various models which we take from Nature, such as Evolutionary Algorithms and Neural Networks. Gene Regulatory Networks are a newer instance of biologically inspired computational models, and so it behooves us to study them further to learn what are the strengths and weaknesses, especially when compared to other bio-inspired models. In this paper, we showed that GRN's can have complex, nonlinear behaviors, which nonetheless can be evolved fairly directly and can be measured using human perception on the combined output of 10's of thousands of artificial cells. GRN's are the most plausible models for dealing with developmental processes, although L-systems, which are closer to symbolic AI, are probably more compact descriptions. Following work in interactive evolution using NEAT and HyperNEAT, we think that GRN's can be as useful, yet more biologically plausible in the natural design of artificial life artifacts, such as robots.

References

- Banzhaf, W. (2003). Artificial regulatory networks and genetic programming. *Genetic Programming Theory and Practice*, pages 43–62.
- Barnsley, M. (1988). *Fractals everywhere*. Academic Press.
- Chavoya, A. and Duthen, Y. (2008). A cell pattern generation model based on an extended artificial regulatory network. *Biosystems*, 94(1-2):95–101.
- Clune, J., Beckmann, B. E., McKinley, P. K., and Ofria, C. (2010). Investigating whether hyperneat produces modular neural networks. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation, GECCO '10*, pages 635–642, New York, NY, USA. ACM.
- Cussat-Blanc, S., Bredeche, N., Luga, H., Duthen, Y., and Schoenauer, M. (2011). Artificial gene regulatory networks and spatial computation: A case study. In *Proceedings of the European Conference on Artificial Life (ECAL'11)*. MIT Press.
- Cussat-Blanc, S. and Pollack, J. (2012). A cell-based developmental model to generate robot morphologies. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*. ACM.
- Davidson, E. H. (2006). The regulatory genome: gene regulatory networks in development and evolution. *Academic Press*.
- Dawkins, R. (1986). The blind watchmaker. *Longman Scientific & Technical*.
- Doursat, R. (2008). Organically grown architectures: Creating decentralized, autonomous systems by embryomorphic engineering. *Organic Computing*, pages 167–200.
- Eggenberger, P. (1997). Evolving morphologies of simulated 3d organisms based on differential gene expression. In *Proceedings of the Fourth European Conference on Artificial Life*, pages 205–213. MIT Press Cambridge, MA.
- Fahlman, S. (1990). The cascade-correlation learning architecture. Technical report, DTIC Document.
- Hart, D. (2007). Toward greater artistic control for interactive evolution of images and animation. *Applications of evolutionary computing*, pages 527–536.
- Joachimczak, M. and Wróbel, B. (2008). Evo-devo in silico: a model of a gene network regulating multicellular development in 3d space with artificial physics. In *Proceedings of the 11th International Conference on Artificial Life*, pages 297–304. MIT Press.
- Joachimczak, M. and Wróbel, B. (2010). Evolving Gene Regulatory Networks for Real Time Control of Foraging Behaviours. In *Proceedings of the 12th International Conference on Artificial Life*.
- Nicolau, M., Schoenauer, M., and Banzhaf, W. (2010). Evolving genes to balance a pole. *Genetic Programming*, pages 196–207.
- Reil, T. (1999). Dynamics of gene expression in an artificial genome - implications for biological and artificial ontogeny. *Advances in Artificial Life*.
- Reynolds, C. (2011). Interactive evolution of camouflage. *Artificial Life*, 17(2):123–136.
- Romero, J. and Machado, P. (2007). *The art of artificial evolution: a handbook on evolutionary art and music*. Springer.
- Secretan, J., Beato, N., D Ambrosio, D. B., Rodriguez, A., Campbell, A., and Stanley, K. O. (2008). Picbreeder: evolving pictures collaboratively online. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, CHI '08*, pages 1759–1768. ACM.
- Sims, K. (1991). Artificial evolution for computer graphics. In *Computer Graphics*.

Finger-painting Fitness Landscapes: An Interactive Tool for Exploring Complex Evolutionary Dynamics.

Luis Zaman^{1,2,3}, Charles Ofria^{1,2}, Richard E. Lenski^{2,3}

¹Department of Computer Science and Engineering

²BEACON Center for the Study of Evolution in Action

³Ecology, Evolutionary Biology & Behavior Program

Michigan State University, East Lansing, MI USA

{zamanlui, ofria, lenski}@msu.edu

Abstract

Evolution involves only a few simple processes, yet the resulting dynamics are surprisingly rich and complex. Sewall Wright developed the metaphor of fitness landscapes to provide deeper insight into the complex workings of evolution. Here we extend that metaphor by visualizing in real time the dynamic processes that drive evolution. We allow viewers to construct fitness landscapes interactively while also varying key parameters including population size, mutation effect size, mode of reproduction (asexual or sexual), and density-dependent selection. This application is both mechanistic and visual, and it thereby allows the active exploration of evolutionary processes. We walk the reader through several exercises including both simple activities potentially suitable for education and examples of deeply conceptual topics that remain the focus of current research in evolutionary biology.

Introduction

Sewall Wright first depicted fitness landscapes in 1932 as a contour plot relating two genetic axes with hills and valleys of fitness (Wright, 1932), and these landscapes are still pervasive in evolutionary biology today. Even at this inception, Wright understood the oversimplifications necessary to depict genetic space in so few dimensions. However, the insights and intuition this visual metaphor has brought to evolutionary thinking are great (Wright, 1988; Gavrillets, 2004). In fact, one of the most important questions in evolution concerns how populations move from local optima to higher (potentially global) optima – a question in which fitness landscapes are central (Pigliucci and Kaplan, 2006). Substantial work has been done to address this question, from appealing to the unintuitive geometry of high-dimensional spaces by Fisher (Whitlock et al., 1995; Orr, 1998) to considering landscapes with mostly neutral mutations by Kimura (1983).

Fitness landscapes play a substantial role in evolutionary computation as well as biology, though the relevance of computational landscapes is less debated because evaluation functions are sufficient to describe fitness surfaces for most optimization problems. For example, Langdon investigated the structure of fitness landscapes for some canonical

evolutionary computation functions such as XOR (Langdon and Poli, 1999; Langdon, 1999). Just as evolutionary biology informs computation, sometimes computation can shed light on biology. Kashtan et al. (2007) showed that changing environments in a digital system gave populations access to peaks they otherwise could not explore. Experiments in Avida, an artificial life platform, demonstrated a “survival of the flattest” effect: at high mutation rates, populations evolved to lower and flatter rather than higher and steeper regions in the fitness landscape (Wilke et al., 2001; Wilke and Adami, 2003).

While the landscape metaphor holds a prominent place in evolutionary thinking, it is not without critics. One criticism concerns the metaphor’s multiple forms: one describing individual fitness as a function of genotypes, another as a function of phenotypes (Simpson, 1953), and yet another describing a population’s mean fitness as a function of its genetic structure (Pigliucci and Kaplan, 2006). Except in a few special cases, the axes are not rigorously defined, but rather depict some sort of distance between types.

A related set of criticisms of the fitness landscape metaphor concern the lack of rigorous mathematical formalism (Provine, 1989). In Wright’s defense, the metaphor was meant to hide the mathematics necessary for describing evolution in massively multi-dimensional spaces, while providing an intuitive framework for considering the various possible outcomes. Gavrillets (2004) distinguishes the mathematical fitness landscape as a high-dimensional formal construct, but he still must show them in two or three dimensions. Rigorous mathematics are necessary for advancing theory in high-dimensional landscapes, but the formalisms may provide little intuition about the evolutionary process. Perhaps this is why, despite the criticisms, depictions of fitness landscapes usually reflect Wright’s original form. Many fundamental concepts in evolution can be illuminated using so simple a metaphor.

Depicting whole populations evolving on fitness landscapes is even more complicated; they are often shown as an abstract cloud moving up a peak. Numerical simulations and other analytical methods are generally required for

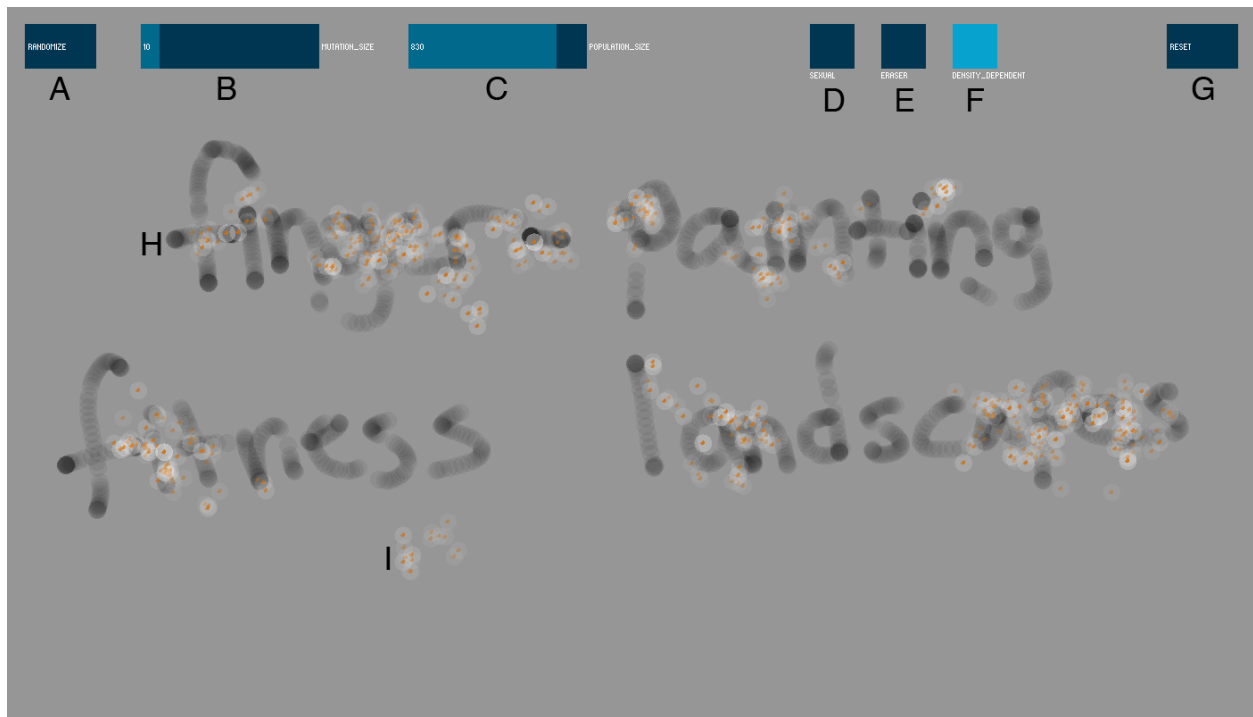


Figure 1: The full screen application view. (A) A button that randomizes each individual's genotype, spreading them out across the fitness surface. (B) A slider that controls the size of mutational effects, which are drawn from a uniform distribution and are applied randomly to one of the two genetic dimensions every time an offspring is produced. (C) A slider that controls population size. (D) An asexual-sexual toggle that, when enabled, causes two parent organisms to produce offspring with an averaged (in both dimensions) genotype for 30% of reproductive events (70% of reproductive events remain asexual). (E) An eraser toggle that switches from the drawing mode to an erasing mode. (F) A toggle for density-dependence, shown enabled, allows the viewer to explore how deformations of the landscape caused by the organisms (e.g., by depleting resources) affect evolutionary dynamics. (G) A reset button that removes all painted regions and randomizes the genotypes. (H) A painted region that represents the fitness surface, where darker areas depict higher fitness. Repeated or slower strokes increase the darkness of touched regions, producing higher fitness peaks. (I) Individuals are depicted as small orange squares, shown here with density-dependence enabled, which suppresses the fitness surface immediately surrounding them, thus lightening the region.

deeper insight about the typical and exceptional paths that populations may take. However, these methods require substantial time and expertise to master, whereas the intuitive finger-painting system that we present provides a simple, fast heuristic tool for interactive discovery. Thus, despite the limitations of Wright's two-dimensional genetic space, it is the most accessible form of fitness landscapes for visualizing the evolutionary process in action and, as such, the representation that we chose to extend. We note, however, that mutations in our system can move genotypes various distances (not single uniform steps) from their progenitors, similar to mutations on phenotypic landscapes.

Interactive System and Touchscreen Display

We built an interactive system that combines visualization and simulation, allowing the user to construct and modify a fitness landscape on which a population evolves in real-time.

Developed for a touchscreen interface, the user can “finger-paint” diverse fitness landscapes. Each stroke slightly darkens the surface, and regions become even darker with additional strokes to the same area. In the visualization, darker regions depict higher fitness levels. The light-gray background regions represent a baseline fitness level, while the maximum fitness level is $\sim 70\%$ higher than the baseline; it requires ~ 100 strokes of a given spot to produce the maximum fitness. These and many other details of the implementation can be changed by modifying the underlying program, but they are not subject to change by the user. However, in addition to finger-painting the fitness landscape, the user can vary several key aspects of the simulation by using toggles and sliders. There is an eraser toggle that, when activated, causes additional touches to restore the corresponding regions to the low baseline fitness. There is also a reset button that allows the user to erase all painted areas. These sim-

ple options let users quickly build complex landscapes that include such important features as hills, valleys, ridges, and plateaus.

In addition to the landscape, the system also displays an evolving population of organisms, with each individual genotype located on the fitness landscape and shown as a semi-transparent orange square. Each genotype has two integer values that provide its coordinates on the drawing surface. The program simulates evolution using continuous rounds of tournament selection. In each round, five individuals are randomly sampled from the population and one individual reproduces with a probability that is determined by its fitness as a proportion of the sum of the five fitness values; each fitness value corresponds to the darkness of the fitness surface at the point where the individual sits. When an individual reproduces, it replaces a randomly chosen organism, thus maintaining a constant population size. The population size can be varied by the user, both before and during a given session, using a slider on the interface.

Offspring are mutated along one randomly chosen dimension, with an offset to the parent's coordinate drawn from a random uniform distribution centered at zero. The range of the distribution is determined by the mutation effect size, which can be adjusted interactively by using another slider. Two additional toggles allow the user to vary reproductive mode (asexual versus sexual) and ecological interactions (negative density-dependent effects). Sexual recombination, when enabled, occurs with a 30% probability at every reproduction event. When recombination is triggered, two parents are chosen by tournament selection, and an offspring is produced by averaging the parents' genotypes in both dimensions. When density-dependent selection is implemented, the fitness of each individual is reduced as it interacts with an increasing number of other individuals. In our simulation, this density-dependence acts over local regions of the fitness landscape rather than globally across the whole population. This local interaction may occur if, for example, different fitness peaks represent different resources that can be drawn down by some genotypes but not others. Thus, the more individuals located in a particular region of the fitness landscape, the lower each individual's fitness will be. We show this dynamic on the screen by lightening the surface (lowering the fitness) in a small circular region around each individual; the surface becomes progressively lighter in regions with higher densities of organisms. We calculate the density of a region using a hidden layer that specifies the radius of density-dependent effects and allows the program to compute quickly the relevant fitness modifiers. Individual fitness is calculated as $(1 - \text{Density}) * \text{PaintedFitness}$, where *Density* is a scaled value between zero and one that represents how depressed the landscape is at a given position. When density-dependence is disabled, *Density* is always set to zero.

There are, of course, important limitations to our sys-

tem, including the representation of all genotypes in a two-dimensional space. In that respect, our system suffers from the same defect as Wright's metaphorical fitness landscapes, as we discussed in the introduction. On the other hand, we have brought this important metaphor to life by allowing the user to paint endless forms of landscapes and then watch the process of evolution in action on the fitness surfaces. Moreover, the user can alter features of the landscape and manipulate key variables even as evolution proceeds.

In the next section, we describe and illustrate several exercises that can be performed using our program. The source code can be downloaded from <http://bit.ly/xn8isR>. Execution requires the Processing Development Environment, which is available from <http://processing.org/>. Additionally, a limited version of the system is viewable in some browsers at <http://bit.ly/zJ7B4N>.

Exercises for the Reader

The finger-painting application is intended to help the user gain intuition about the dynamics of evolution on fitness landscapes. To that end, we outline below four "exercises for the reader" that span a wide range of evolutionary principles. We begin with depictions of two basic and well-known concepts perhaps appropriate for educational activities: the hill-climbing process driven by natural selection; and the potential for random drift to allow small populations to cross fitness valleys and thereby discover other nearby fitness peaks. We then present two more exercises that illustrate areas of active research: the role of density-dependent effects in flattening the fitness landscape and thus promoting diversity; and how high mutation pressure can favor organisms that occupy flatter, rather than higher, regions of the fitness landscape.

Hill Climbing

Natural selection reflects disproportionate reproduction by *individuals* with high fitness. In the context of fitness landscapes, natural selection is often described as a hill-climbing process, whereby the *population* moves from regions of lower to higher fitness. Despite its intuitive simplicity for those familiar with the basic ideas, there are confusing aspects of the hill-climbing metaphor, especially the important distinction between the unguided behavior of individuals and the systematic advance of the entire population up a local fitness peak. By seeing the process of individuals producing more or fewer offspring based on their fitness levels, and the resulting hill-climbing effect in the population, the user may develop a mechanistic understanding of evolution by natural selection.

To illustrate this process, start by gently touching the screen to create a low (light gray) peak on the fitness surface, as shown in Figure 2 A. After the population has converged on this peak (pressing the Randomize button on the screen will re-disperse the population if necessary), begin drawing

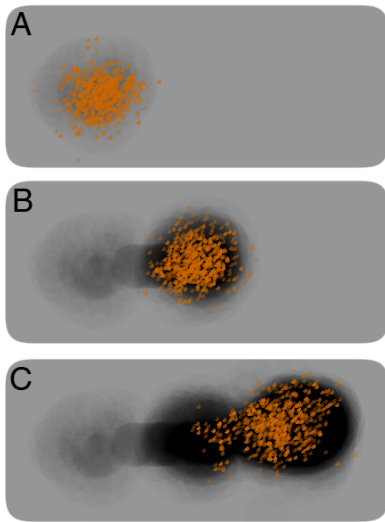


Figure 2: Demonstration of hill-climbing dynamic. (A) Begin with the population converged on a fairly low fitness peak. (B and C) Add progressively darker regions of higher fitness, taking care so that these new peaks are not separated by a wide fitness valley, and watch the population move up-hill as more and more individuals occupy the higher regions of the landscape.

a progressively darker region adjacent to the initial peak. If the regions overlap, you will see some new individuals - the product of reproduction and mutation - near the new peak. And because the individuals that are nearer to the new peak have higher fitness, they will reproduce more offspring, so that the population as a whole climbs toward the new peak. If the initial low peak and new higher peak do not overlap, then you can draw a bridge of intermediate fitness that connects them, as shown in Figure 2 B. You can continue this process by drawing additional nearby peaks that are progressively darker and thereby observe the hill-climbing dynamics of an evolving population, as shown in Figure 2 C.

You might then re-start the simulation by pressing the reset button. After proceeding as before through one or two rounds of building adjacent peaks, you can then draw an even higher peak but at a large distance from the other peaks (not shown). You should see that the population does not immediately (if ever) climb that distant peak, despite its high fitness. The different population behavior with respect to connected and disconnected regions of the fitness landscapes shows that evolution finds local fitness peaks more readily than global ones.

Small Populations and Drift

Evolution involves the interplay of several underlying processes. Natural selection reflects the differences in *expected* reproductive success of individuals based on their genotypes

and their fit to the environment. In the context of our application, an individual's expected reproductive success is proportional to the darkness of its location in the fitness landscape. But each individual's *realized* reproductive success also depends on chance. In our application, the tournament selection probabilistically favors more fit individuals but does not guarantee that the most fit will reproduce and, moreover, any individual may be eliminated at random whenever another individual reproduces. In evolutionary parlance, these random aspects of survival and reproduction are called genetic drift. In large populations, the fluctuations caused by genetic drift are relatively small and tend to be overwhelmed by the systematic hill-climbing effect of natural selection. In small populations, however, these random fluctuations can be more important. Of particular interest here, individuals with lower fitness (off the current peak) may replace those of higher fitness (on the current peak). This process reduces the population's mean fitness, but it sometimes also allows the population to cross a fitness valley and discover another, possibly higher, fitness peak (Whitlock, 1995). This effect of small population size was a central part of Wright's Shifting Balance Theory (Wright, 1932, 1982), in which random genetic drift allows populations to move between fitness peaks.

To see this effect, start by drawing a single fitness peak of moderate height (darkness) and allow a large population to converge on it. Set the mutation effect size to be very small (between 5 and 10), and then draw a second higher (darker) peak that is separated from the first peak by a narrow valley, as shown in Figure 3 A. Notice that this large population stays centered on the first, lower peak because there is selection against genotypes in the low-fitness valley. Now lower the population size to about 10 or 20 individuals and observe how the population becomes much more dynamic, in the sense that its center of mass frequently wanders away from the center of the first peak (Fig. 3 B). The population will occasionally even fall off the peak, so that several individuals can be found in the fitness valley between the two peaks. After some time, the population may move onto the second peak, having crossed the valley that was impassable by the larger population.

Density-dependence and Diversity

In the previous exercises, most or all individuals ended up in one region of the fitness landscape, which means there was very little genetic diversity. But the biological world is incredibly diverse, so we would like to understand how evolution produces and sustains that diversity. There are many factors that affect biological diversity, and in this exercise we will demonstrate one important factor that concerns the nature of interactions among organisms. Density-dependence refers to biological processes for which the rates depend on the density of organisms. For example, in the familiar model of logistic population growth, the per capita

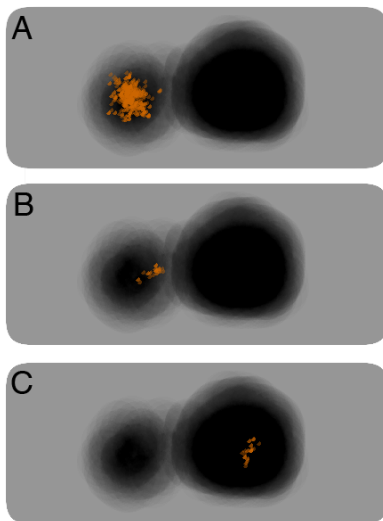


Figure 3: Demonstration of the effect of genetic drift in small populations. Start with a large population that has already converged on a fitness peak. Then add a second, higher peak that is nearby but separated from the first peak by a valley, so that the population does not exhibit the hill-climbing behavior shown in Figure 2. (A) In the large population, new genotypes that are off the peak are quickly replaced by more fit individuals. (B) Now reduce the population size to a very few individuals. New genotypes that are off the peak are not replaced as quickly and some may fall into the basin of attraction of the second peak. (C) After the second peak has been colonized, the population will then typically exhibit the familiar hill-climbing dynamic.

rate of reproduction declines as the population density increases. Density-dependent selection refers to situations in which the fitness of genotypes depends on the number of interactions between individuals. In the case of negative density-dependence, the fitness of an individual declines when it has more interactions with other individuals. (Frequency-dependent selection is a similar concept. Because population density is constant in our application, except when changed by the user, frequency-dependent and density-dependent effects are equivalent.) Negative density-dependent effects often result from increased competition for resources, but they can also result from interactions with predators or parasites whose density increases with that of their prey or hosts. In the context of fitness landscapes, we expect these negative interactions to be more intense among similar genotypes than among those that are dissimilar. In this exercise, we show how that variation in interaction strength promotes diversity by allowing subpopulations to coexist on multiple fitness peaks.

For this exercise, begin by drawing two adjacent fitness peaks of different height. Make sure that the population

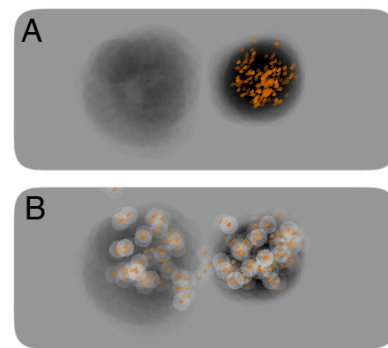


Figure 4: Demonstration of negative density-dependent selection and its effect on diversity. Draw two nearby but distinct fitness peaks of unequal height. Set both population and mutation effect sizes to intermediate or high values, then randomize the population. (A) With density-dependence turned off, the entire population will converge on the higher peak. (B) Now activate the toggle for density-dependence, and observe how the population spreads out and occupies both peaks.

size and mutation effect size are both restored to intermediate or high values (not kept at the small values from the previous exercise). After randomizing the population, the vast majority of individuals will soon occupy only the higher (darker) peak, as seen in Figure 4 A, because individuals on the higher peak produce more offspring than those on the lower peak. Now activate density-dependence and observe that a lighter region surrounds each individual. The lighter color indicates a depression in the fitness landscape relative to the level if that individual were not there. Notice, too, that this effect increases when multiple individuals are in close proximity. Now watch as the population spreads out, first over the current peak and then onto the second peak, as illustrated in Figure 4 B. This shift occurs because the individuals on the first peak depress their own fitness to the point that the second peak becomes the higher one. The two subpopulations - species, perhaps - will then coexist indefinitely.

Survival of the Flattest at High Mutation Rates

Evolution is often described colloquially as survival of the *fittest*. That is, genotypes with high fitness tend to produce more offspring and thereby propel the population up a local peak, as we saw in the first exercise. However, if the peak is very narrow and mutation effects are large, then high-fitness individuals tend to produce offspring that have fallen off the peak and thus have low fitness. In that case, selection may favor genotypes that are *less* fit, in the sense of producing fewer offspring, but more robust because mutations tend to have less harmful effects on their offspring. This scenario has been dubbed “survival of the flattest” because the more

robust types occupy lower but flatter regions of the fitness landscape rather than high but narrow peaks (Wilke et al., 2001). This phenomenon is thought to be important in both computational and biological systems (Wilke et al., 2001; Wilke and Adami, 2003; Beardmore et al., 2011).

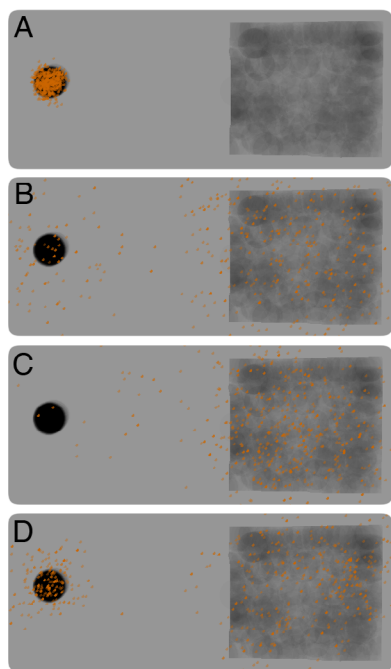


Figure 5: Demonstration of survival of the flattest. (A) Start with a high but narrow fitness peak, and allow the population to converge on it. Set the mutation effect size to a low value. Now draw a second peak that is much lower and broader than the first peak. (B, C) Gradually raise the mutation effect size, and watch as the population moves to the second flatter peak. (D) Now suddenly reduce the mutation effect size to a low value, and the population may move back to the high but narrow fitness peak.

Survival of the flattest is easy to demonstrate, even though the effect is a recent discovery. Figure 5 shows the setup, with a single high but narrow fitness peak, and a much broader but lower peak or plateau. When the mutation effect size is small, the population will remain tightly centered on the high but narrow peak (Figure 5 A). But as you gradually raise the mutation effect size, notice that the population becomes a progressively larger cloud, with many low-fitness offspring born off the peak (Figure 5 B). As you increase the mutation effect size even more, the population abandons the high but narrow peak entirely, and spreads out across the lower, flatter peak (Figure 5 C). If the mutation rate is suddenly reduced back to a low level, the population may shift back to the high but narrow peak (Figure 5 D), although this reversal also depends on the distance between the two peaks in relation to

other parameters.

Further Explorations

In all of the previous exercises, the mode of reproduction was asexual, which is the default when one begins the application. The interested reader might want to repeat the previous exercises, except with sexual reproduction enabled using the toggle on the display screen. In what cases are the outcomes similar for asexual and sexual reproduction, and when do they differ? We would suggest, in particular, that readers explore the effects of reproductive mode in combination with density-dependent effects. We observed before that density-dependent interactions induced asexual populations to diversify and thereby occupy multiple peaks, as though the subpopulations had split into distinct species. With sexual reproduction, however, intermediate forms (hybrids) are continually generated. To explore the consequences, the reader can switch back and forth between asexual and sexual modes of reproduction, add and remove peaks, and so on.

Conclusions

We built an interactive visualization system that allows users to create fitness landscapes by finger-painting them on a blank canvas. By doing so, Wright's largely metaphorical fitness landscape becomes a playground where one can hone intuition for more formal future experimentation and analysis. Our system is effective for building intuition because all of the processes are visual and mechanistic, while the entire process can be watched in real-time. In addition to painting the initial landscape, users can interact with the system by adding or erasing fitness peaks and by changing parameters such as mutation effect size. We outlined several examples that span a range of complexity from educational exercises to actively researched topics.

Acknowledgments

This work benefited greatly from discussions with Brian Connelly, Justin Meyer, and many other members of the BEACON Center for the Study of Evolution in Action. This material is based in part upon work supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

- Beardmore, R. E., Gudelj, I., Lipson, D. A., and Hurst, L. D. (2011). Metabolic trade-offs and the maintenance of the fittest and the flattest. *Nature*, 472:342–346.
- Gavrilets, S. (2004). *Fitness Landscapes and the Origin of Species*. Princeton University Press, Princeton, N.J.

- Kashtan, N., Noor, E., and Alon, U. (2007). Varying environments can speed up evolution. *Proceedings of the National Academy of Sciences*, 104:13711–13716.
- Kimura, M. (1983). *The Neutral Theory of Molecular Evolution*. Cambridge University Press, New York, N.Y.
- Langdon, W. (1999). Scaling of program fitness spaces. *Evolutionary Computation*, 7:399–428.
- Langdon, W. and Poli, R. (1999). Boolean functions fitness spaces. *Genetic Programming*, pages 651–652.
- Orr, H. (1998). The population genetics of adaptation: the distribution of factors fixed during adaptive evolution. *Evolution*, pages 935–949.
- Pigliucci, M. and Kaplan, J. M. (2006). *Making Sense of Evolution: The Conceptual Foundations of Evolutionary Biology*. University of Chicago Press, Chicago.
- Provine, W. (1989). *Sewall Wright and Evolutionary Biology*. University of Chicago Press.
- Simpson, G. G. (1953). *The Major Features of Evolution*. Columbia University Press, New York, N.Y.
- Whitlock, M. C. (1995). Variance-induced peak shifts. *Evolution*, 49:252–259.
- Whitlock, M. C., Phillips, P. C., Moore, F. B.-G., and Tonsor, S. J. (1995). Multiple fitness peaks and epistasis. *Annual Review of Ecology and Systematics*, 26:601–629.
- Wilke, C. O. and Adami, C. (2003). Evolution of mutational robustness. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, 522:3–11.
- Wilke, C. O., Wang, J. L., Ofria, C., Lenski, R. E., and Adami, C. (2001). Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, 412:331–333.
- Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding and selection in evolution. In *Proceedings of the 6th International Congress of Genetics*, volume 1, pages 356–366.
- Wright, S. (1982). The shifting balance theory and macroevolution. *Annual Review of Genetics*, 16:1–20.
- Wright, S. (1988). Surfaces of selective value revisited. *The American Naturalist*, 131:115–123.

Evolution in Action

Extended Abstracts

The Influence of Genetic Operators and their Probabilities on the Behavior of Lizards within the Calangos Game

Diego J. D. Almeida¹, Emanuel M. C. Tavares¹, Venyton N. L. Izidoro¹, Leandro N. de Castro¹,
Angelo C. Loula², Charbel N. El-Hani³

¹Natural Computing Laboratory - Mackenzie Presbyterian University - São Paulo, Brazil

²Cognitive and Intelligent Systems Lab - State University of Feira de Santana - Feira de Santana, Brazil

³History, Philosophy and Biology Teaching Laboratory - Federal University of Bahia - Salvador, Brazil
{venyton.izidoro, djduarte, emanuelmaues}@gmail.com, lnunes@mackenzie.br,
angelocl@ecom.ufes.br, charbel.elhani@pq.cnpq.br

Extended Abstract

The Calangos Game is based on the modeling of a real ecological case about lizards that inhabit a desert-like field of sand dunes in the middle San Francisco River, located in the Caatinga biome, in Bahia, Brazil (Rocha et al., 2005). The project aims at developing an electronic game to aid in the teaching and learning of ecology and evolution (Loula, et al., 2009; Oliveira, et al., 2009; Oliveira, et al., 2010).

The goal of this paper is to investigate the influence of genetic operators and their probabilities in a simulator of the Calangos game. Based on these results it will be possible to choose what types of crossover operators and probabilities will be embedded in the game. The idea is to use an evolutionary algorithm to embody evolution within the lizards. To do so, a proper genetic representation and operators were designed, as summarized in Table 1.

Table 1. Lizard's Genotype.

Feature	Domain	Feature	Domain
Gender (<i>Sx</i>)	{female, male}	Preference for Fruits (<i>Fp</i>)	10 – <i>Ip</i>
Body size (<i>Bs</i>)	{10.0cm, 30.0cm}	Circadian activity cycle (<i>Cc</i>)	{daytime, nighttime}
Head width (<i>Hs</i>)	[<i>Bs</i> /5 – <i>Bs</i> /10, <i>Bs</i> /5 + <i>Bs</i> /10]	Ability to bury in the sand (<i>Sa</i>)	{yes, no}
Color Pattern (<i>Pc</i>)	{visible, stealthy}	Preferential temperature (<i>Tmp</i>)	{cold, warm, hot}
Maximum Velocity (<i>Vm</i>)	(<i>Bs</i> /10) + [1.0, 5.0]	Minimum hydration threshold (<i>Th</i>)	[20%, 50%]
Sociability (<i>Agr</i>)	{yes, no}	Minimum energy threshold (<i>Te</i>)	[20%, 50%]
Preference for Insects (<i>Ip</i>)	[0.0, 10.0]	Preference for Fruits (<i>Fp</i>)	10 – <i>Ip</i>

Four types of crossover operators were evaluated (*single-point*, *n-point*, *uniform* and *weighted average*) and the *random resetting* mutation (Back et al., 2000). A simulator was implemented (Izidoro et al., 2011) and the results include *longevity* (*L*), *fecundity* (*F*) and *fitness* ($fit = F + 0.01 * L$).

For each combination of mutation rate, crossover operator and rate, 10 tests were performed and the average results were taken, as shown in Table 2.

Table 2. Best results of all combinations evaluated. From each combination of operators and values, those that resulted in the best values (longevity, fecundity and fitness) were detached in gray, and the best values of all are in bold.

Crossover Operator	Mutation Rate	Crossover Rate	Longevity±(std)	Fecundity±(std)	Fitness	
					Best--Worst	Average±(std)
Single-Point	10%	10%	1092±(677,91)	21,96±(7,68)	153,15--11,97	55,18±(21,75)
		30%	1231,83±(591,4)	22,43±(8,57)	160,23--12,35	57,55±(22,35)
	20%	10%	1163,45±(890,61)	18,23±(10,29)	117,71--15,25	48,36±(28,97)
N-Point	1%	30%	1159,71±(474,42)	22,13±(8,65)	149,83--13,33	56,18±(21,76)
		30%	1181,12±(582,41)	23,07±(7,67)	167,88--11,59	58,32±(20,79)
	10%	60%	1246,05±(596,87)	20,37±(8,6)	143,52--12,47	53,52±(22,63)
		20%	1223,41±(489,32)	24,29±(8,45)	166,27--15,3	61,16±(21,36)
Uniform	5%	30%	1174,2±(713,67)	19,57±(10,05)	128,83--14	51,19±(26,6)
		30%	1136,73±(480,21)	24,24±(7,15)	174,73--11,12	60,22±(18,86)
	20%	30%	902,31±(821,56)	16,02±(10,25)	107,88--11,44	41,31±(28,21)
Weighted Average	5%	30%	902,31±(821,56)	16,02±(10,25)	107,88--11,44	41,31±(28,21)
	20%	30%	957,11±(643,71)	16,71±(9,21)	110,23--10,1	43,26±(24,25)

In general terms, the best performance was obtained with the n-point crossover and, thus, this was selected to be implemented in the game. Future work include investigation into the influence of the parameters and operators on the survival strategies of the lizards, the main causes of death, and the performance of the different operators and probabilities in hostile and friendly environments.

Acknowledgements

The authors thank MackPesquisa, CNPq and Fapesp for the financial support.

References

- Back, T., Fogel, D. B., and Michalewicz, Z. (2000). *Evolutionary Computation: Basic Algorithms and Operators*. Institute of Physics Publishing.
- Rocha, P. L. B., and Rodrigues, M. T. (2005). Electivities and Resource Use by an Assemblage of Lizards Endemic to the Dunes of The São Francisco River, Northeastern Brazil. *Pap. Avulsos de Zoologia*, **45**(22):261-284.
- Izidoro, V. N. L., de Castro, L. N., Loula, A. C., and El-Hani, C. N. (2011). The Environmental Influence on the Players' Strategy in a Simulation of the Ecological and Evolutionary Level of the Game, *Proc. of the Brazilian Symposium on Games and Digital Entertainment (SBGames)*.
- Loula, A. C., Munoz, Y. J., Vargens, M. M. F., Apolinário, A. L., de Castro, L. N., Rocha, P. L. B., and El-Hani, C. N. (2009). Modelagem Ambiental em um Jogo Eletrônico Educativo. *Proc. of the Brazilian Symposium on Games and Digital Entertainment (SBGames)*, pages 171-180.
- Oliveira, E. S., Apolinário, A. L., and Loula, A. C. (2009). Simulação Física - Ambiental e Interface Gráfica em Um Jogo Eletrônico Educacional. *Proc. of the IX Regional Computing School Bahia, Sergipe and Alagoas (ERBASE)*.
- Oliveira, E. S., Calmon, J., Apolinário, A. L., Loula, A. C. (2010). Desenvolvimento de Personagens para um Jogo Eletrônico de Ensino e Aprendizagem de Biologia. *Proc. of the X Regional Computing School Bahia, Sergipe and Alagoas (ERBASE)*.

Evolution of migratory-like behavior in Avidians.

Francis Bartlett, Fred Dyer and Robert Pennock.

Michigan State University, United States

bartle47@msu.edu

Extended Abstract

There is a strong interest in navigation behavior in both animals and machines (see Breed & Moore, 2012 for literature review on non-human animals). The majority of this work is concerned with the algorithms and physiological mechanisms that best perform particular navigation tasks. To date there are very few attempts to study how complex navigation systems evolve. It is certain that sophisticated movement strategies such as path integration arose in steps from simpler precursors. Since it is very difficult to study these transitions in evolved biological systems, very few hypotheses exist to explain the evolutionary path of such behavior. Our work investigates the evolution of sequential navigation behavior in Avida. Our methodology deviates from the standard Locic-9 environment (Lenski et al., 1999) to support sensing and movement of the digital organisms. We included additional instructions; move, rotate, sense and sense direction (Grabowski et al., 2010). We limited the sensing ability such that organisms could only “see” the state of their current position, with no ability to look ahead. Their behavioral environment, called the state grid, was a 25x50 rectangular lattice with a simple resource setup (Fig 1). The northern half of the grid provided energy while the southern half

supported reproduction, with neither half allowing for both. Organisms lived and reproduced in a separate 60 x 60 population grid. This allowed the Avidians to compete for space and cycling time in the population grid while moving around in the state grid without complications that arise from navigational interactions (collisions, obstructions etc.) The arrangement of our environment was inspired by shorelines where aquatic organisms venture onto land to forage but otherwise live and reproduce in water. The structure of this task is also similar to lifetime migratory routes of many biological species. The geography and reward structure of both grids remained unchanged throughout each evolutionary run. We hypothesized that this simple task would require the evolution of directed-movement and may resemble the very early conditions under which current navigation strategies evolved.

After 60,000 generations of evolution using this set up, Avidians

Figure 1. Grid set up.

learned that the ability to simply move fast and collect resource was sufficient for survival in this environment.

In order to explore more sophisticated strategies, we systematically changed the environment to discourage wall crawling by making the borders and diagonals a “no-man’s-land” where organisms could neither reproduce or collect resource (Fig 3). In addition we randomized the orientation and location of each organism’s starting position on the grid throughout evolution. We seeded these runs with a wall-crawling ancestor and after 60,000 generations the dominant organisms in the population possessed migration-like behavior. The most complicated strategies took advantage of the sense of direction and used tight sensory input to motor output loops. One of the most striking results was that in six separate evolutionary episodes (identical except for the random seed) all of the final dominant organisms timed some portion

of their behavior by monitoring the growth of their offspring. This regulation of behavior used the if-label instruction which governs the execution or termination of a behavioral loop by checking whether or not a combination of non-operation instructions have been copied to the offspring’s genome. In all cases, this strategy controlled the timing of resource collection. From a biological perspective this is analogous to evolved life history decisions where living organisms must optimize their timing of reproduction.

Finally, we tested the flexibility of the six dominant organisms with round-robin tournaments in the environment of evolution as well as a series of transfer environments where the

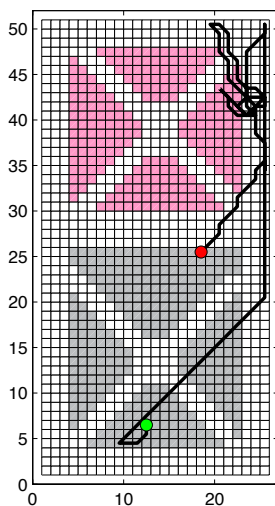


Figure 3. Evolved organism with directed movement behavior.

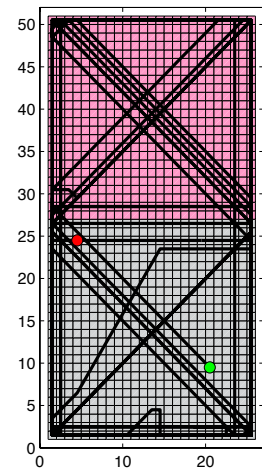


Figure 2. Cockroach-like wall crawling behavior.

distribution of resource and reproductive zones differed from the original environment (Fig 4). The most successful organisms in the evolved environment also performed the best in the majority of the transfer competitions. This was a surprising result since one might expect that the most efficient navigators from the environment of evolution would be very tightly adapted to the original environment and perhaps fare poorly in the transfer scenarios.

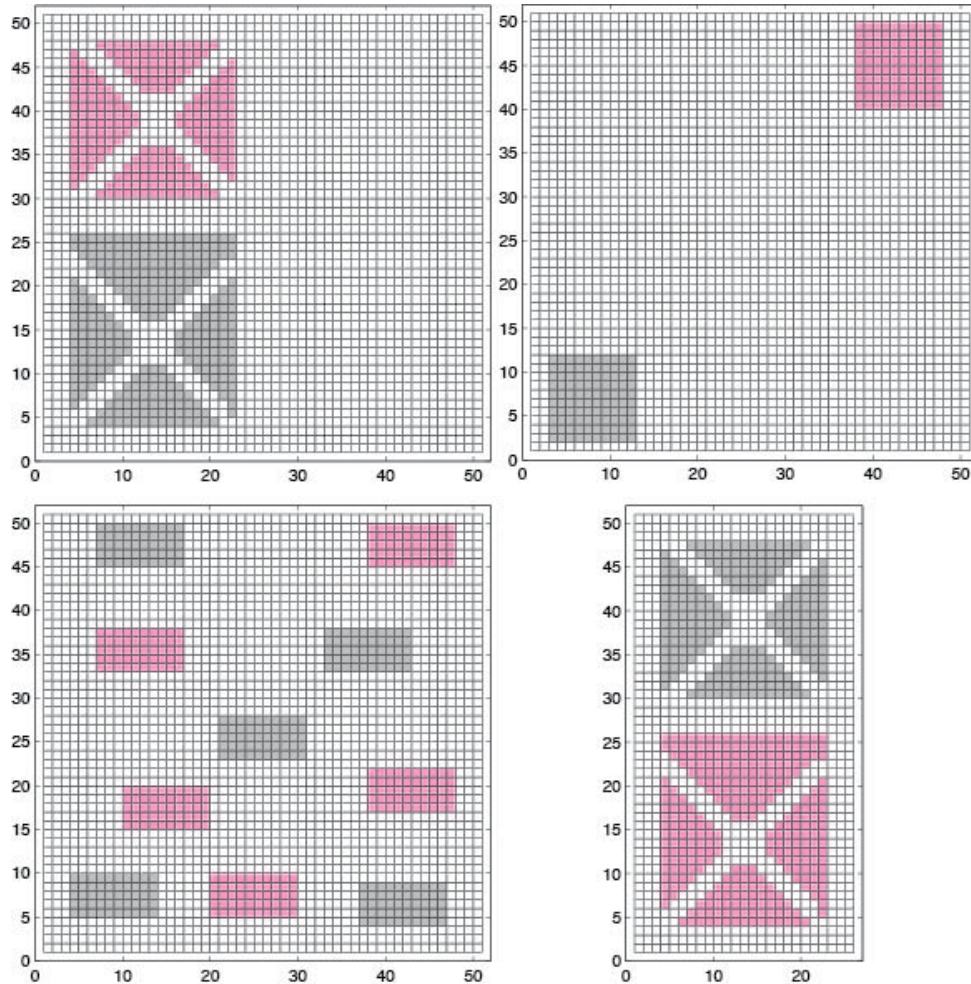


Figure 4. Transfer environments. Pink remained the resource zone and grey the reproductive zone during all transfer tests.

References

- Breed, M and Moore, J. editors (2010). *Encyclopedia of Animal Behavior*. Elsevier. Ltd.
- Grabowski, LM. Bryson, DM. Dyer, F. Pennock, RT. and Ofria, C. (2010). Early Evolution of Memory Usage in Digital Organisms. Proc of the 12th International Conference on Artificial Life. Pgs 224-231. Odense, Denmark.
- Lenski, RE. Ofria, C. Collier, TC, and Adami, C. (1999). Genome complexity, robustness and genetic interactions in digital organisms. *Nature*. Pgs 661-664.

Evidence of Speciation in an Experimental Population of *E. coli* Following the Evolution of a Key Adaptation

Zachary D. Blount^{1,2} and Richard E. Lenski^{1,2}

¹Department of Microbiology and Molecular Genetics, Michigan State University, East Lansing, MI, USA

²BEACON Center for the Study of Evolution in Action, East Lansing, MI, USA

blountza@msu.edu

Extended Abstract

Speciation is among the most fundamental and pervasive of evolutionary processes across all forms of life (Darwin 1859, Mayr 1963, Coyne and Orr 2004, Ptacek and Hankison 2009). However, many questions about the process and pattern of speciation remain unanswered because speciation is rarely “caught in the act” (Coyne and Orr 2004). Progress in understanding speciation as a general phenomenon – one applicable to both natural and artificial life – would benefit from a tractable model system in which speciation could be examined from initial divergence through the later stages of the process. Here we present evidence that a lineage expressing a key innovation that evolved in an experimental population of *Escherichia coli* has become a new species and is amenable for use as a speciation model system.

The Long-term Evolution Experiment (LTEE) was begun in 1988, when 12 *E. coli* populations were founded from a single clone. These populations have since evolved for more than 50,000 generations in a glucose-limited medium called DM25 (Lenski et al 1991, 2004). DM25 also contains citrate, a potential second resource that *E. coli* cannot grow on under the oxygen-rich conditions of the experiment (Scheutz and Strockbine 2005). After 31,000 generations, a variant capable of aerobic citrate utilization (Cit⁺) arose in one of the twelve populations (Blount et al 2008).

Several lines of evidence suggest that the Cit⁺ lineage qualifies as a new species under various species concepts. The Cit⁺ trait transcends the accepted range of variation for *E. coli*, which is partly defined as a phenotypic species by its Cit⁻ phenotype (Scheutz and Strockbine 2005). Cohan’s Ecotype Species Concept (ESC) emphasizes irreversible ecological divergence consequent to niche discovery mutations (NDMs) that move a lineage into a new niche where it can then undergo sweeps of beneficial mutations independent of the parent population. This ecological and evolutionary independence allows the new lineage to coexist with its parent population while continuing to adapt and diverge (Cohan and Perry 2007). The new Cit⁺ trait clearly involved one or more NDMs that gave access to a previously unexploited niche. Moreover, while the Cit⁺ subpopulation eventually rose to numerical dominance, a Cit⁻ subpopulation persisted through at least 40,000 generations. Extensive whole-genome sequencing confirmed that the Cit⁻ and Cit⁺ subpopulations are phylogenetically distinct (Fig. 1). The Cit⁺ lineage is therefore also a new species under the ESC.

Mayr’s Biological Species Concept (BSC) equates speciation with the evolution of reproductive barriers between sexually-reproducing lineages (Mayr 1963). Although bacteria are asexual, genetic exchange via horizontal gene transfer mechanisms is possible, suggesting a means of applying the BSC to bacteria. Under this approach, speciation is evidenced by niche-specific adaptive mutations (NSAMs) that improve a divergent lineage’s fitness in its new niche while reducing its fitness in the ancestral niche. The beneficial fitness effects of NSAMs are expected to be specific to the genetic background in which they arise, and should therefore reduce hybrid fitness following recombination between diverging lineages. Consequently, NSAMs will produce a barrier to successful genetic exchange between diverging lineages that is analogous to reproductive isolation in the BSC. Between 31,500 and 40,000 generations, a period of marked improvement in growth on citrate, the Cit⁺ lineage experienced a dramatic and progressive decline in fitness in the ancestral glucose niche (Fig. 2). This finding indicates that the Cit⁺ lineage has accumulated NSAMs that our future work will seek to identify and evaluate. Overall, our results suggest that Cit⁺ may well be a new, laboratory-evolved species by criteria that satisfy the ESC and BSC, as well as the phenotypic species concept. Given the tractability of *E. coli* in general, the Cit⁺ lineage in particular has utility as a model system with which to study speciation “in action”, especially as regards the genetics of speciation and the formation of barriers to gene exchange.

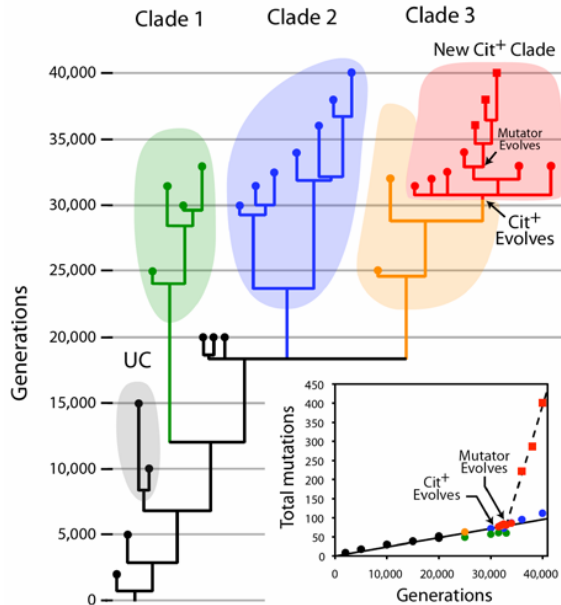


Fig. 1 | Phylogeny of population that evolved Cit⁺. Symbols at branch tips indicate placement of 29 clones isolated at different times and subjected to whole-genome sequencing. Shaded areas and colored symbols identify major clades. Inset shows total number of mutations relative to the ancestor. Note that the Cit⁺ lineage late evolved a mutator phenotype and then subsequently accumulated mutations much more rapidly.

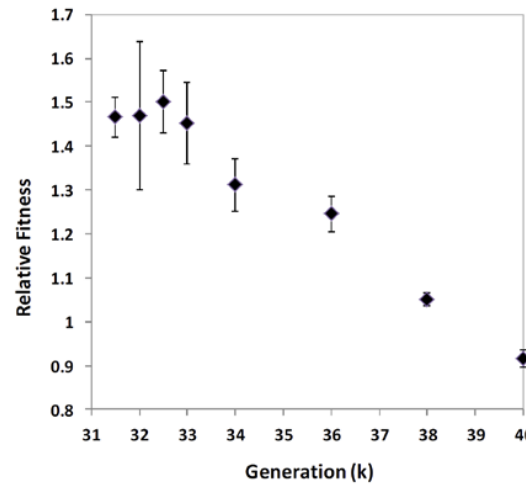


Fig. 2 | The Cit⁺ lineage loses fitness in the DM25 glucose niche over time. Cit⁺ clones were isolated at 8 times. Cit⁻ mutants were derived from each clone. These mutants were then competed against the ancestral strain in DM25. Fitness of each mutant relative to the ancestor was determined as described in Lenski et al (1991). Error bars are 95% confidence intervals.

References

- Blount, Z.D., Borland, C.Z., and Lenski, R.E. (2008). Historical contingency and the evolution of a key innovation in an experimental population of *Escherichia coli*. *Proceedings of the National Academy of Sciences (USA)*, 103:7899–7906.
- Cohan, F.M. and Perry, E.B. (2007). A systematics for discovering the fundamental units of bacterial diversity. *Current Biology* 17:R373–R386.
- Coyne, J.A. and Orr, H.A. (2004). *Speciation*. Sinauer Associates, Sunderland, MA.
- Darwin, C.R. (1859). *On the Origin of Species*. John Murray, London.
- Lenski, R.E., Rose, M.R., Simpson, S.C., and Tadler, S.C. (1991). Long-term experimental evolution in *Escherichia coli*. I. Adaptation and divergence during 2,000 generations. *The American Naturalist*, 138:1315–1341.
- Lenski, R.E. (2004). Phenotypic and genomic evolution during a 20,000-generation experiment with the bacterium *Escherichia coli*. *Plant Breeding Reviews* 24:225–265.
- Mayr, E. (1963). *Animal Species and Evolution*. Belknap Press, Cambridge, MA.
- Ptacek, M.B. and Hankison, S.J. (2009). The pattern and process of speciation. In Ruse, M. and Travis, J., editors, *Evolution: The First Four Billion Years*, pages 177–207. Belknap Press, Cambridge, MA.
- Scheutz, F., and Strockbine, N.A. (2005). Genus I. *Escherichia*, Castellani and Chalmers 1919. In Garrity, G.M., Brenner, D., Kreig, N.R., Staley, F.R. *Bergey's Manual of Systematic Bacteriology, Volume Two: The Proteobacteria*, pages 607–624. Springer, New York, NY.

Cooperation and antagonism in information exchange between two species

Andrés C. Burgos and Daniel Polani

Adaptive Systems Research Group, University of Hertfordshire, Hatfield, UK
d.polani@herts.ac.uk

Extended Abstract

Introduction

We consider the problem of two species which have the option of exchanging information about their environment, thereby improving their chances of survival. For this purpose, we model a system consisting of two species whose dynamics in the world will be model by a Kelly gambling-like strategy which tries to “bet” on future environmental conditions. It is well known that such models lend themselves to elegant information-theoretical interpretations by relating their respective growth rate to the information the individual species has about its environment.

We are specifically interested in modeling how these dynamics are affected when the species interact cooperatively or in an antagonistic way against each other. For this purpose, we consider information exchange between the two species in the framework of an informational “quasi-game”. The latter, while sharing some structural similarities with traditional game theory, constitutes a distinct concept with some properties peculiar to information theory that we introduce here to study the conditions for informational cooperation or antagonism to appear.

Model

Our system consists of two populations X and Y which are modeled by *logistic maps* and resources R such that:

$$\begin{cases} X_{t+1} = r_x X_t (1 - X_t) \\ Y_{t+1} = r_y Y_t (1 - Y_t) \\ R_{t+1} = r_R R_t - \gamma(X_t + Y_t) \end{cases} \quad (1)$$

where X_t and Y_t represent the population density of species X and Y at time t , respectively, and density is the ratio of the existing population to the carrying capacity, which in our case is 1. R_t corresponds to the resources available at time t , growing by a constant rate r_R and used up with a metabolic rate γ by X and Y .

Growth rate

As introduction, let us first consider the simple case where we have a single population X living in an environment. We will represent the environment as a discrete random variable

E with some probability distribution p , where each event represents certain environmental conditions. We define the strategy a species X follows in environment E as a probability distribution π_x over E . We can interpret the probability $\pi_x(e')$ as a proportion of the population prepared for environment e' . If e' does not match the true environment e at the next time-step, we assume that this proportion of the population will die out completely. The proportion of the population that matched the environment grows at a given rate. Thus, the total growth rate of the population when environment e occurs is given by

$$r_x = \sum_{e'} \pi_x(E = e') f(e, e')_{R_t, X_t, Y_t} \quad (2)$$

where f is the reproduction rate, such that $f(e, e')_{R_t, X_t, Y_t} = \delta_{ee'} g(R_t, X_t, Y_t)$, and g is a sigmoid function depending on the resources after consumption.

From this definition we can see that populations prepared only to single environmental conditions will inevitably die out, given that conditions change with time. Therefore, populations will be selected to ‘hedge their bets’, trying to maximize their expected or long-term growth rate.

Expected growth rate We define the expected growth rate of population X in environment E following a strategy π_x as the expected value of the logarithm of the growth rate in a single generation

$$W(E) = \sum_e p(E = e) \log \pi_x(E = e) f(e, e)_{R, X, Y} \quad (3)$$

since $f(e, e')_{R, X, Y} = 0$ when $e \neq e'$.

Optimal expected growth rate The question that arises now is: how should a population be prepared for future environmental conditions in order to maximize their expected growth rate?

The solution is given by a strategy called *proportional betting* or Kelly-gambling (J. L. Kelly, 1956). It says that for a population to maximize its expected growth rate, it should

“bet” (be prepared) on each possible environment proportionally to its probability of occurring, *i.e.* $\pi_x = p$. We can now define the optimal expected growth rate of a population X in environment E as

$$W^*(E) = \max_{\pi_x} W(E) = F - H(E) \quad (4)$$

where $F = \sum_e p(E = e) \log f(e, e)_{R,X,Y}$.

Notice here that f must have the form of a diagonal matrix for proportional betting to be optimal. From equation 4 we can make two interesting statements: our strategy is independent of the reproduction rate f and we achieve the maximum expected growth rate when environmental uncertainty is eliminated.

Optimal expected growth rate with side information

We consider now the same population X but with the ability of acquiring information from the environment through sensors S_x . The environment affects the state of the sensors. We assume that the amount of information they can acquire at each time-step is fixed in time. We know that an informational cue increases the expected growth rate exactly by an amount equal to the mutual information between the cue and the environment (Donaldson-Matasci et al., 2010). Then the optimal expected growth rate with sensor information is

$$W^*(E|S_x) = W^*(E) + I(E : S_x) \quad (5)$$

Two species exchanging information We now add to our model a new species Y with the same characteristics as X , and consider for both species a way of communicating with each other. An important assumption is that the amount of information a population will communicate will be proportional to its density. We assume that the bigger a population is, the more information it will release to the other. We model environment, sensors and communication as random variables. The relationships between these are modeled by a Bayesian network, as shown in Figure 1.

$$C_x \leftarrow S_x \leftarrow E \longrightarrow S_y \longrightarrow C_y$$

Figure 1: Bayesian network

The optimal expected growth rate of species X with sensors S_x and receiving information C_y is

$$W^*(E|S_x, C_y) = W^*(E|S_x) + I(E : C_y|S_x) \quad (6)$$

The relation between all the defined optimal expected growth rates for species X (analogously for Y) is the following:

$$W^*(E) \leq W^*(E|S_x) \leq W^*(E|S_x, C_y) \leq F \quad (7)$$

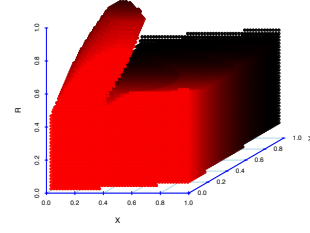


Figure 2: Concealing volume for species X . The plot is better appreciated in colors.

Results

We studied the dynamics of the system through computer simulations in the framework of an informational “quasi-game”. As stated before, a species in our model will share an amount of information proportional to its density. We now consider species that decide whether they share this information or not. The dynamics in our model is that of a quasi-game, as mixed strategies in our scenario do not lead to linear mixing of utilities as in traditional games.

We analyse the equilibrium strategies (sharing or concealing information) for the resulting payoffs expressed in terms of the expected growth rate of a species looking two steps ahead for different initial values of X , Y and R .

The volume shown in the plot corresponds to a concealing strategy, which turns out to be strictly dominant. This volume corresponds to situations where few resources are left after two time-steps. Particularly, the salient volume represents situations where the sum of population densities after the first time-step is near the amount of available resources, resulting in a high resource consumption even if there was a high initial amount of it.

Conclusion

We have considered two species exchanging information in an environment with limited resources. We have seen that side information about the environment is translated into an increase in the expected growth rate of the species. With these few assumptions, we were able to identify the conditions for which concealing information is expected to emerge. Namely, hiding information from the others is an optimal strategy when resources are scarce.

References

- Donaldson-Matasci, M. C., Bergstrom, C. T., and Lachmann, M. (2010). The fitness value of information. *Oikos*, 119:219–230.
- J. L. Kelly, J. (1956). A new interpretation of information rate. *Bell System Technical Journal*, 35:917–926.

The Evolution of Modularity Under Changing Environments in Digital Organisms

Rosangela Canino-Koning and Charles Ofria

BEACON Center for the Study of Evolution in Action
Department of Computer Science and Engineering
Michigan State University, East Lansing, MI 48824 USA
caninoko@msu.edu

Extended Abstract

Genetic modularity contributes to both the robustness and evolvability of natural genomes by allowing functional components to evolve independently of each other, and allowing the formation of robust gene regulatory networks (Hartwell et. al., 1999) (Kirschner and Gerhart, 1998). However, the evolutionary origins of biological modularity are not well understood. Modularity appears to be a costly adaptation. Under constant environmental conditions, computational models of biological evolution do not produce modular structures. Rather, existing modular structures tend to evolve away (Kashtan and Ulon, 2005). However, changing environments with modular goals have been shown to promote the evolution of modular structures in neural networks (Kashtan and Ulon, 2005).

We used the Avida Digital Evolution Platform (Ofria and Wilke, 2004) to examine the effects of changing environments on the genomes of evolving digital organisms. Digital organisms are self-replicating computer programs with linear genomes, which are well suited to examine whether the principles established in prior studies apply in a true a-life context.

In constant environments, where environmental conditions do not change over time, there is significant selective pressure to overlap and condense genetic components that share portions of their function. This functional overlap streamlines the execution of those components, giving the organisms a competitive advantage over their less efficient peers. However, these condensed genomes, because of their tight structure and high pleiotropy, tend to be less evolvable than less tightly condensed genomes (Waxman and Peck, 1998). However, in long-cycle fluctuating environments, where environmental conditions cycle between different states over a long period of time, there arises a new selective pressure to evolve genomes that are more evolvable. These genomes can thus more easily gain or lose functions as dictated by environmental conditions (Kirschner and Gerhart, 1998).

In order to examine the effects of these competing selective pressures, we subjected a total of 150 independent populations of digital organisms to two types of two-phase cyclical changing environments: a benign changing environment, and a hostile changing environment, plus a constant (non-changing) environment as a control.

In the benign changing environment, the organisms were rewarded for performing a specific logical operation (the fluctuating task) during the first phase, and then not rewarded for performing the task during the second phase. In the hostile changing environment, the organisms were rewarded during the first phase, and then punished for performing the fluctuating task during the second phase. In the control, the fluctuating task was continually rewarded, regardless of phase. In all treatments, the organisms were also continually rewarded for performing a separate logical operation (the backbone task) (Figure 1).

Under these conditions, we showed that the genetic architectures evolved in response to changing environments differed significantly from those evolved in the constant environment. Specifically, in all treatments, the genomes combined and overlapped most of the instructions responsible for both the backbone and

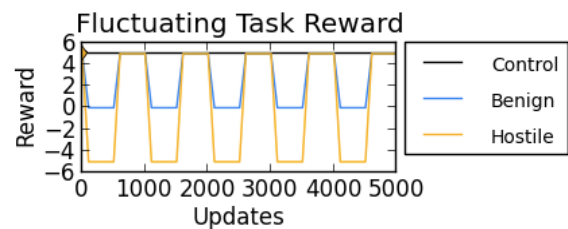


Figure 1 - Fluctuating Task Reward in Control, Benign and Hostile Changing Environments

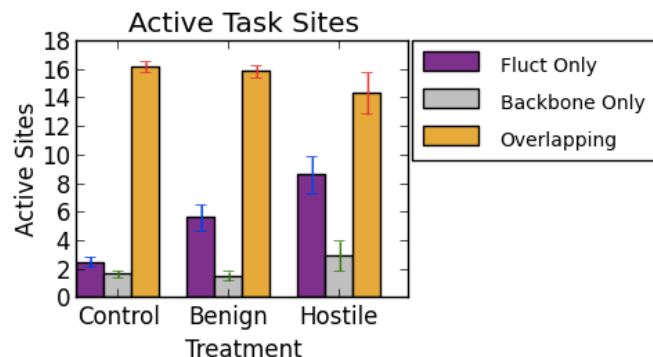


Figure 2 - Active Task Sites of the Final Dominant Genotypes

fluctuating tasks, but the genomes evolved in the changing environments reserved significantly more of the non-overlapping portions of the instructions responsible for performing the fluctuating task in separate sections of their genomes (Figure 2 – Fluct Only). These separate sections are able to evolve independently of the overlapping core functional regions (Figure 2 – Overlapping), thus preserving the integrity of the backbone task and significant portions of the fluctuating task through environmental changes. As a result, this organizational motif dramatically improves the ability of the genomes to use small mutations to “switch on and off” the performance of the fluctuating task in response to changes in the environment.

References

- L. Hartwell, J. Hopfield, and S. Leibler, “From molecular to modular cell biology,” *Nature*, 1999.
- N. Kashtan and U. Alon, “Spontaneous evolution of modularity and network motifs,” *Proceedings of the National Academy of Sciences*, vol. 102, no. 39, p. 13773, 2005.
- M. Kirschner and J. Gerhart, “Evolvability,” *Proceedings of the National Academy of Sciences*, vol. 95, no. 15, pp. 8420–8427, 1998.
- C. Ofria and C. O. Wilke, “Avida: A Software Platform for Research in Computational Evolutionary Biology,” *Artificial Life*, vol. 10, pp. 191-229, 2004.
- D. Waxman and J. R. Peck, “Pleiotropy and the Preservation of Perfection,” *Science*, vol. 279, no. 5354, pp. 1210–1213, Feb. 1998.

Evolution of Aging and Rejuvenation in Bacteria

Lin Chao, Camilla U. Rang and Annie Y. Peng

University of California San Diego
LChao@ucsd.edu

Extended Abstract

Single-celled organisms dividing by binary fission were thought not to age. A study by Stewart *et al.* (2005) reversed the dogma by demonstrating that *Escherichia coli* were susceptible to aging. Stewart *et al.* and others have shown that bacteria age because a dividing mother cell partitions non-genetic damage (e.g. oxidized proteins) asymmetrically to her two daughters. Thus, bacteria that divide symmetrically partition damage asymmetrically. However, a follow-up study by Wang *et al.* (2010) countered those results by demonstrating that *E. coli* cells trapped in microfluidic devices are able to sustain robust growth without aging. The present study reanalyzed these conflicting data by applying a population genetic model for aging in bacteria (Chao 2010). The model allows for the asymmetrical partitioning of damage, which provides a fitness advantage by increasing variance and the efficiency of natural selection. Our reanalysis of the data (Rang, Peng and Chao, 2011) showed that in *E. coli*, as predicted by the model, (1) aging and rejuvenation occurred simultaneously in a population; (2) lineages receiving sequentially the maternal old pole converged to a stable attractor state; (3) lineages receiving sequentially the maternal new pole converged to an equivalent but separate attractor state; (4) cells at the old pole attractor had a longer doubling time than ones at the new pole attractor; and (5) the robust growth state identified by Wang *et al.* corresponds to our predicted attractor for lineages harboring the maternal old pole. Thus, the previous data, rather than opposing each other, together provide strong evidence for bacterial aging. Outcomes (1) - (4), as predicted by the model, are illustrated graphically in Fig. 1.

The evolution of bacterial aging driven by the asymmetrical partitioning of non-genetic damage has broad implications. Traditional evolutionary theory has long postulated that life history and/or soma and germ line are required for the evolution of aging (Turke 2008). The evolution of damage asymmetry in *E. coli* argues that they are not. If the first single-celled organisms partitioned damaged symmetrically, they would have been selected for asymmetry as soon as damage rates increased. Because daughters produced by symmetrical partitioning are identical, there would have been no life history before asymmetry. However, with the evolution of asymmetry, life history would have emerged. Moreover, if the daughter that receives more damage is regarded as a continuation of the mother, then the larger damage fraction can be interpreted to be equivalent to soma. Both are a component of the phenotype that is kept by the mother and not transmitted to the new daughter. Thus, aging, life history, soma, or at least its functional equivalence, all emerged simultaneously with asymmetry and may be as ancient as the first cell.

Could artificial life evolve aging? Focusing only on those that can evolve, one can imagine that they could. If life history already existed, they could evolve aging by traditional evolutionary mechanisms in which better early reproduction exacted a cost or tradeoff later in life. However, could aging in artificial life evolve by the same manner we are observing in bacteria? Our work points to the importance of non-genetic damage. Thus, if the genotype to phenotype map was perfect in artificial life, and thus lack a non-genetic component, the evolution of aging would not be possible by the process we propose. However, if artificial life could acquire a deleterious non-genetic component to its phenotype, we would predict that it would.

References

- Chao, L. (2010). A Model for Damage Load and Its Implications for the Evolution of Bacterial Aging. *PLoS Genet* 6, e1001076.
 Rang, Camilla U., Peng, Annie Y. and Chao, L. (2011). Temporal Dynamics of Bacterial Aging and Rejuvenation. *Current biology*, 21:1813-1816.
 Stewart, E.J., Madden, R., Paul, G., and Taddei, F. (2005). Aging and death in an organism that reproduces by morphologically symmetric division. *PLoS Biology* 3:295-300.
 Turke, P.W. (2008). *Williams's theory of the evolution of senescence: Still useful at fifty*. *Quarterly Review of Biology*, 83:243-256.
 Wang, P., Robert, L., Pelletier, J., Dang, W.L., Taddei, F., Wright, A., and Jun, S. (2010). Robust Growth of *Escherichia coli*. *Current Biology*, 20:1099-1103.

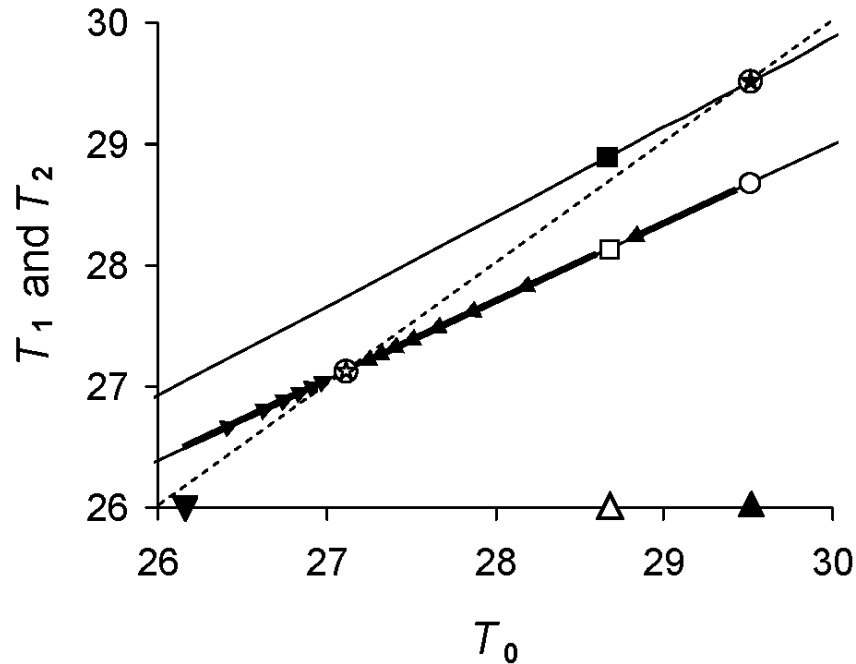


Figure. 1. Graphical prediction for aging and rejuvenation in bacteria. Solid black and open symbols refer respectively to old and new poles in this and subsequent figures. Predictions based on model of Chao (2010) and are presented on a phase plot of T_1 and T_2 as a function of T_0 . T_1 and T_2 are the doubling times of daughters 1 and 2, which receive respectively the new and old poles of the mother cell. T_0 is the doubling time of the mother cell. The predicted relationships are represented as two solid lines (lower, T_1 ; upper, T_2). Dashed line (---) is the identity line. The intercept of the identity line and the T_1 and T_2 graph lines corresponds to attractor states to which the new and old pole lineages converge (\otimes , \otimes). To illustrate the convergence, consider a mother cell with a doubling time of 29.5 min. Projecting upwards from $T_0=29.5$ (\blacktriangle) to the two solid lines identifies the two points \circ and \otimes , which are the predicted T_1 and T_2 values for her two daughters. We denote \circ to be \tilde{T}_1 . Following only the new pole lineage, let the \tilde{T}_1 become a mother by projecting her doubling time onto the T_0 axis (Δ). \tilde{T}_1 's daughters will have the doubling times T' and T'' (\square , \blacksquare). By tracking \tilde{T}_1 and T' , the new pole lineage progresses downwards along the T_1 graph line (right most arrow). Because \tilde{T}_1 is greater than T' , the progression corresponds to rejuvenation. If the T' daughter and her subsequent daughters are likewise projected in turn onto the T_0 axis as mothers, the resulting new pole lineage converges to the lower attractor (\otimes). If the initial mother cell had a $T_0=26.2$ min (\blacktriangledown), a similar convergence occurs, albeit from the left of the attractor (left most arrows), and the increase in doubling times corresponds to aging. Note that the old pole graph line also has its own attractor point (\otimes). Figure reproduced from Rang *et al.* (2011).

Specialization by *Burkholderia cenocepacia* Biofilm Ecotypes Limits Adaptation in a Planktonic Environment

Crystal N. Ellis¹, Rachel K. Staples² and Vaughn S. Cooper²

¹Massachusetts General Hospital, GRJ520, 55 Fruit St., Boston, MA 02114

²University of New Hampshire, Rudman Hall, 26 College Rd., Durham, NH 03824
vaughn.cooper@unh.edu

Extended Abstract

An organism's potential to adapt to new environments depends on access to beneficial mutations, and access is sometimes restricted by adaptive history, which refers to the course of genetic and ecological events that contributed to adaptation in a prior niche (Travisano et al., 1995; Bennett et al., 1992; Dykhuizen and Davies, 1980; Zhong et al., 2009; Crill et al., 2000; Caley and Munday, 2003). Experimental evolution of microbial populations illustrates how access to mutations is affected by adaptive history and ecological role (Shade et al., 2011; Lenski et al., 2006; Masel et al., 2007; Baquero, 2009; Trindade et al., 2009; Khan et al., 2011; Woods et al., 2011). Understanding adaptability and the influence of ecology in complex bacterial communities is essential given that growth in multispecies biofilms may be the predominant mode of microbial life. We examined the adaptability of Early (~315 generation) and Late (~1050 generation) ecotypes that evolved in a previously described *B. cenocepacia* biofilm population (Poltak and Cooper, 2010). Populations founded by a single clone underwent selection in a daily cycle of surface colonization, biofilm formation, and dispersal for ~1,050 generations in test tubes containing a polystyrene bead. Each population evolved increased biofilm production and diversified into ecologically, genetically, and morphologically distinct types: two specialists, Wrinkly (W) and Rough (R), which despite slower growth rates produce the most biofilm, and a Studded generalist (S), which grows fast and produces little biofilm (Poltak and Cooper, 2010). We hypothesized that the specialists' adaptive histories would limit adaptation in a planktonic environment; we investigated this by experimentally evolving generalist and specialist clones in planktonic culture for an additional 300 generations. Adaptability was quantified as 1) the time required for a population to generate and become dominated by an optimal phenotype (Quayle and Bullock, 2006), 2) the time to extinction of the ancestor (Grimm and Wissel, 2004), 3) the planktonic fitness of evolved mutants, and 4) the tradeoffs experienced by the evolved mutants. We also identified the biochemical and genetic mechanisms underlying the magnitude of each ecotype's adaptability, and described potential mechanisms for reversion from a biofilm to a planktonic lifestyle. Most evolved populations converged on a dominant S mutant regardless of the adaptive history of the

ancestral founder (Fig. 1A). However, the S mutants evolved from generalists were more fit than those evolved from specialists when each were competed against the wild-type clone ($t = 4.43$, $df = 4$, $p = 0.01$). Additionally, mutants of Early ancestors remained fit under biofilm conditions, whereas those from Late ancestors experienced tradeoffs ($F = 94.06$, $df = 7$, $p = 8.45 \times 10^{-12}$). Generalists seemed more adaptable than specialists (Fig. 1B), and long periods of biofilm specialization produced low fitness in the planktonic environment at the expense of ancestral, biofilm fitness (Fig. 1C). The rate at which the S mutant appeared was also influenced by the founding genotype, suggesting adaptation was slowed by negative epistasis between the supply of mutations conferring a benefit in the planktonic environment and those acquired during biofilm adaptation. Since the ancestral Early and Late W specialists are genotypically distinct, we also determined if planktonic adaptation proceeded along different pathways related to the genotype. The S mutants evolved from W specialists acquired unique mutations, however, each also shared a mutation in a gene affecting sugar metabolism, which may be the source of increased fitness in planktonic culture. In summary, the adaptability of biofilm mutants can be described as a function of prior specialization, owing to the magnitude of specialization, genetic tradeoffs of new mutations, and epistatic interactions with prior mutations.

References

- Baquero, F. (2009). Environmental stress and evolvability in microbial systems. *Clinical Microbiology and Infection*, 15: Suppl.: 1.
- Bennett, A., Lenski, R., and Mittler, J. (1992). Evolutionary adaptation to temperature. I. Fitness responses of *Escherichia coli* to changes in its thermal environment. *Evolution*, 46: 16-30.
- Caley, J., and Munday, P. (2003). Growth trades off with habitat specialization. *Proceedings of the Royal Society of London, Series B*, 270: S175-S177.
- Crill, W., Wichman, H., and Bull, J. (2000). Evolutionary reversals during viral adaptation to alternating hosts. *Genetics*, 154: 27-37.
- Dykhuizen, D. and Davies, M. (1980). An experimental model: bacterial specialists and generalists competing in chemostats. *Ecology*, 61: 1213-1227.
- Grimm, V. and Wissel, C. (2004). The intrinsic mean time to extinction: a unifying approach to analyzing persistence and viability of populations. *Oikos*, 105: 501-511.

A.

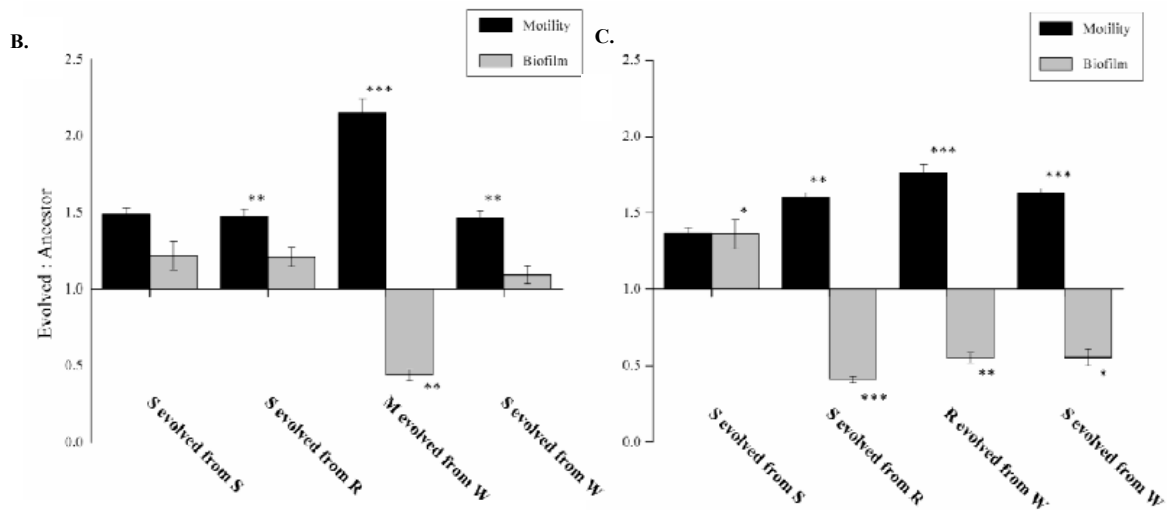
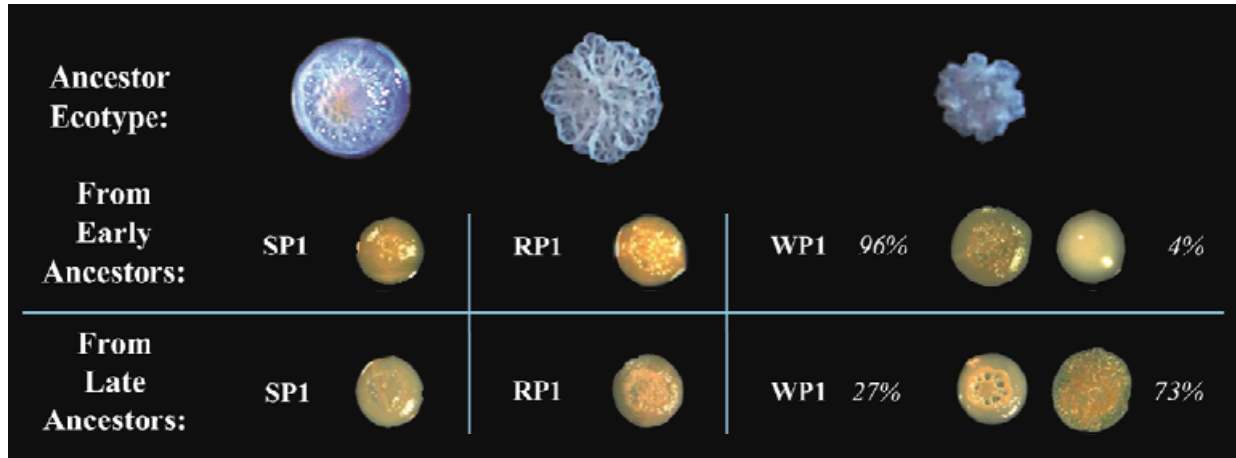


Figure 1: A) Colony morphologies of variants from populations evolved by planktonic serial passage for 300 generations. The ancestral ecotypes (top) from the Early and Late biofilm-adapted populations were each used to found three replicate populations for serial transfer, resulting in 18 total planktonic populations (representative populations are shown here). All derived populations were monocultures containing studded mutants with the exception of populations founded by wrinkly, which consisted of two mutants. Percentages indicate the frequency of each colony type in the population. B) Biofilm production (Gray) and motility (black) of planktonic-evolved variants represented as a ratio against their Early founding ancestors and C) Late founding ancestors. S = Studded, R = Rough, W = Wrinkly, and M = Mucoid. Bars below 1.0 on the X axis represent standard error (df = 7). P values are the results of Tukey's HSD tests: * $p < 0.001$, ** $p < 1e10^{-5}$, *** $p < 1e10^{-9}$.

Khan, A., Dinh, D., Schneider, D., Lenski, R. and Cooper, T. (2011). Negative epistasis between beneficial mutations in an evolving bacterial population. *Science*, 332: 1193-1196.

Lenski, R., Barrick, J., and Ofria, C. (2006). Balancing robustness and evolvability. *PLoS Biology*, 4: e428.

Masel, J., King, O., and Maughan, H. (2007). The loss of adaptive plasticity during long periods of environmental stasis. *American Naturalist*, 169: 38-46.

Poltak, S. and Cooper, V. (2010). Ecological succession in long-term experimentally evolved biofilms produces synergistic communities. *The ISME Journal*, 5: 369-378.

Quayle, A. and Bullock, S. (2006). Modeling the evolution of genetic regulatory networks. *Journal of Theoretical Biology*, 238: 737-753.

Shade, A., Read, J., Welkie, D., Kratz, T., Wu, C., and McMahon, K. (2011). Resistance, resilience and recovery: aquatic bacterial

dynamics after water column disturbance. *Environmental Microbiology*, 13: 2752-2762.

Travisano, M., Mongold, J., Bennett, A., and Lenski, R. (1995). Experimental tests of the roles of adaptation, chance and history in evolution. *Science*, 6: 87-90.

Trindade, S., Sousa, A., Xavier, K., Dionisio, F., Ferreira, M., and Gordo, I. (2009). Positive epistasis drives the acquisition of multidrug resistance. *PLoS Genetics*, 5: e1000578.

Woods, R., Barrick, J., Cooper, T., Shrestha, U., Kauth, M., and Lenski, R. (2011). Second-order selection for evolvability in a large *Escherichia coli* population. *Science*, 331: 1433-1436.

Zhong, S., Miller, S., Dykhuizen, D., and Dean, A. (2009). Transcription, translation, and the evolution of specialists and generalists. *Molecular Biology and Evolution*, 26: 2661-2678.

Experimental evolution of an artificial bacterial mutualism

Kazufumi Hosoda¹, Akihiro Asao², Shingo Suzuki¹ and Tetsuya Yomo^{1,2,3}

¹Graduate School of Information Science and Technology, Osaka University, Yamadaoka 1-5, Suita, Osaka, 565-0871, Japan

²Graduate School of Frontier Bioscience, Osaka University, Yamadaoka 1-3, Suita, Osaka, 565-0871, Japan

³Exploratory Research for Advanced Technology, Yamadaoka 1-5, Suita, Osaka, 565-0871, Japan
yomo@ist.osaka-u.ac.jp

Extended Abstract

Organisms are totally symbiotic in nature (Douglas 1994). As natural selection is not simple in symbiosis, the evolution of symbiosis is still unclear. Especially mutualism, an interaction between species that is beneficial to both, is known to be evolutionarily problematic in its development and maintenance, because the mutualism is based on the interaction which does not necessarily provide direct benefit to the own fitness (Sachs and Simms 2006). Various theoretical studies have provided general understandings regarding the natural selection (ultimate factor) in the evolution of mutualisms or cooperation systems (Hamilton 1964, Foster and Wenseleers 2006, Nowak 2006). However, it is very challenging to gain a general understanding regarding the physical/structural property of organisms that affects the evolution, because the natural mutualisms are already well-developed and those histories are extremely complex.

Insight can be gained by not only retracing the history of natural symbiosis, but also by observing the experimental evolution (Lenski 1992) of an artificial symbiosis which is composed of previously independent populations (Momeni et al. 2011). Artificial symbioses make it possible to directly observe the evolutionary trajectory from its origin and to analyze the detailed changes in the system. Moreover, we can reconstitute unrealistic situations such as where the original one species is mixed with the evolved partner species. Studies using artificial mutualisms have already shown that constituent organisms increase their growth properties (*i.e.*, develop the mutualism) in the experimental evolution despite their intrinsic evolutionary problem (Shendure et al. 2005, Shou, Ram and Vilar 2007, Hillesland and Stahl 2010). On the other hand, other artificial systems clearly show the evolutionary vulnerability of the mutualism and cooperation system by adding engineered ‘defectors’ (Griffin, West and Buckling 2004, Harcombe 2010). Thus, organisms might rarely become fatal defectors without engineering and seem to favor mutualism or cooperation as a whole. We believe the existence of some general physical/structural properties of organisms that support mutualisms under the principle of natural selection, and detailed analyses of such experimental evolutions of artificial mutualisms will achieve to capture them.

Previously, we constructed an artificial mutualism composed of two genetically engineered auxotrophic strains of *Escherichia coli*: one strain lacked a gene necessary for the Ile biosynthesis and the other lacked that for Leu, designated as Γ and L^- , respectively (Hosoda et al. 2011, Hosoda and Yomo 2011) (Fig. 1A). This is one of the simplest mutualism and the pair of Ile and Leu is one of the best pairs in ~1000 pairs tested before using 46 auxotrophic types (Wintermute and Silver 2010). In minimal medium without amino acid supplements, both strains did not grow in the monoculture but grew in the coculture; therefore, the mutualism is obligate relationship. We found that L^- cells, ~10 h after mixing with Γ cells before its own population growth, began to oversupply Ile ~50-fold greater than in the monoculture, eventually leading to continual growth of both strains. Thus, this previous study shows a quick adaptation to the emergence of a nascent mutualism.

In this study, we investigated the evolution of the same mutualism. First, we examined whether the mutualism develop in the experimental evolution (Fig. 1B). Specifically, we cocultured at three different initial cell concentrations (high, mid, low). The growth rate of the cells in this mutualism depends on the cell concentration (higher concentration shows faster growth). We evaluated the growth rate of the total cell population in each culture, and selected a culture with the lowest initial concentration among the cultures whose growth rate were greater than a threshold criterion. Then we transferred the selected culture to the next three cultures in such a way that the mid initial concentration of the next cultures equaled the initial concentration of the previously selected culture. Therefore, the initial concentration should decrease over transfer if the mutualism develops, and increase if not. As a result, the initial concentration decreased over transfer (Fig. 1C; ~10 generations per transfer). Thus, our artificial mutualism developed as well as other artificial mutualisms in the previous studies, and it became ready for further analyses. Our experimental system is very simple and makes us easily possible to comprehensively analyze the evolution by such as genome, transcriptome, and metabolome analysis, directing us to finding the general physical/structural properties of organisms that bring symbiotic nature.

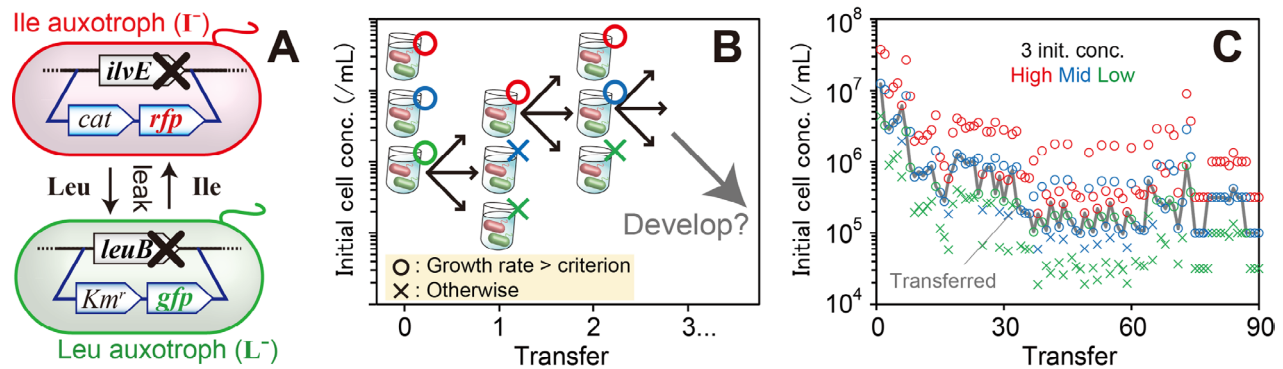


Figure 1: (A) Schematic of our artificial mutualism. (B) Schematic of the rule in culture transfer. (C) Results of the experimental evolution.

References

- Douglas, A. E. 1994. *Symbiotic interactions*. Oxford ; New York: Oxford University Press.
- Foster, K. R. & T. Wenseleers (2006) A general model for the evolution of mutualisms. *Journal of Evolutionary Biology*, 19, 1283-1293.
- Griffin, A. S., S. A. West & A. Buckling (2004) Cooperation and competition in pathogenic bacteria. *Nature*, 430, 1024-7.
- Hamilton, W. D. (1964) The genetical evolution of social behaviour. I. *J Theor Biol*, 7, 1-16.
- Harcombe, W. (2010) Novel cooperation experimentally evolved between species. *Evolution*, 64, 2166-2172.
- Hillesland, K. L. & D. A. Stahl (2010) Rapid evolution of stability and productivity at the origin of a microbial mutualism. *Proc Natl Acad Sci U S A*, 107, 2124-2129.
- Hosoda, K., S. Suzuki, Y. Yamauchi, Y. Shiroguchi, A. Kashiwagi, N. Ono, K. Mori & T. Yomo (2011) Cooperative adaptation to establishment of a synthetic bacterial mutualism. *PLoS One*, 6, e17105.
- Hosoda, K. & T. Yomo (2011) Designing symbiosis. *Bioeng Bugs*, 2, 338-41.
- Lenski, R. E. 1992. Evolution, Experimental. In *Encyclopedia of Microbiology*, ed. J. Lederberg, 125-140. San Diego: Academic Press.
- Momeni, B., C. C. Chen, K. L. Hillesland, A. Waite & W. Shou (2011) Using artificial systems to explore the ecology and evolution of symbioses. *Cell Mol Life Sci*, 68, 1353-68.
- Nowak, M. A. (2006) Five rules for the evolution of cooperation. *Science*, 314, 1560-3.
- Sachs, J. L. & E. L. Simms (2006) Pathways to mutualism breakdown. *Trends Ecol Evol*, 21, 585-92.
- Shendure, J., G. J. Porreca, N. B. Reppas, X. Lin, J. P. McCutcheon, A. M. Rosenbaum, M. D. Wang, K. Zhang, R. D. Mitra & G. M. Church (2005) Accurate multiplex polony sequencing of an evolved bacterial genome. *Science*, 309, 1728-32.
- Shou, W., S. Ram & J. M. Vilar (2007) Synthetic cooperation in engineered yeast populations. *Proc Natl Acad Sci U S A*, 104, 1877-82.
- Wintermute, E. H. & P. A. Silver (2010) Emergent cooperation in microbial metabolism. *Mol Syst Biol*, 6.

Darwinian Evolution of Translation-coupled RNA Self-replication System

Norikazu Ichihashi^{1,2}, Kimihito Usui¹, Yasuaki Kazuta¹, and Tetsuya Yomo^{1,2,3}

¹ JST ERATO

² Graduate School of Information Science and Technology, Osaka University

³ Graduate School of Frontier Bioscience, Osaka University

yomo@ist.osaka-u.ac.jp

Extended Abstract

Darwinian evolution, the consequence of variation and natural selection, has played a central role in the emergence of the characteristics of life. Because of the acquisition of the ability to undergo Darwinian evolution, primitive life, which had been simple and inefficient, has gradually become more complex and sophisticated. Although this scenario is widely accepted, it remains purely speculative. Here, we attempted to construct an evolvable reaction system from non-living molecules and observe the evolutionary process directly.

For molecules to evolve, they must have three capabilities: self-replication, variation, and heredity. To construct these abilities, we used an RNA encoding an RNA replicase and a reconstituted cell-free translation system based on the previous RNA replication system (Mills et al. (1967)). Consequently, the RNA replicase was translated from the RNA, and the replicase replicated the original RNA (Fig. 1, Kita et al. (2008)). To link the genotype to the phenotype, we encapsulated the reaction into a water-in-oil emulsion. Theoretically, repeating this self-replication reaction would produce various RNA mutants by replication error, and consequently a mutant RNA with better replication ability would dominate the population by natural selection.

First, we repeated the RNA self-replication reaction with supporting amplification process include reverse-transcription, PCR, in vitro transcription, and encapsulation with fresh translation system. After 60 h of cumulative incubation, the average self-replication activity of the RNA increased more than 10-fold. Second, we repeated the RNA self-replication with simpler procedure includes fusion with new emulsions containing fresh translation system and division of emulsion by filtration. After 200 h of cumulative incubation, the average self-replication activity further increased about 100-fold. Sequence analysis revealed that about 40 mutations had been introduced into the RNA during the cycle of replication and most of them had become fixed in the population. These data demonstrated that RNA with greater replication capability spontaneously evolved in our system. Kinetic analysis of RNA clones revealed that many parameters have changed after the evolution, especially the parameters involved in resistance of parasitic replicator, which spontaneously appears during the reaction probably through RNA recombination (Bansho et al. (2012)). This result indicates that the self-replicating RNA acquired tolerance to parasitic replicator by Darwinian evolution.

Parasitic replicator has been considered as one of the major hindrances for the emergence of sustainable self-replication system. Our result provides an experimental evidence for the potential of RNA and protein self-replication system to produce a parasite-resistant self-replication mechanism. Direct observation of the evolutionary process as described here would provide experimental insights to gain a better understanding of the evolution of life.

References

- Mills, D.R. et al. (1967). An extracellular Darwinian experiment with a self-duplicating nucleic acid molecule. *Proc Natl Acad Sci USA* 58, 217-224.
- Kita, H. et al. (2008). Replication of genetic information with self-encoded replicase in liposomes. *ChemBiochem*, 9:2403–24106.
- Bansho Y. et al. (in press). Importance of parasite RNA species repression for prolonged translation-coupled RNA self-replication. To appear in the *Chemistry and Biology*.

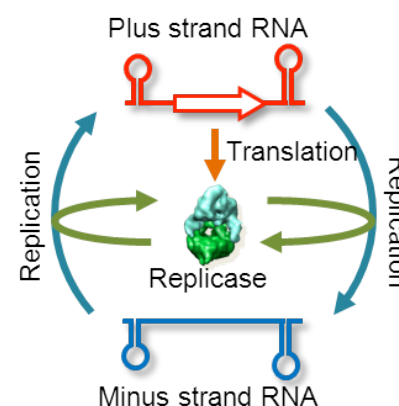


Fig. 1 Translation-coupled RNA self-replication system

Network Representation of the Game of Life and Self-organized Criticality

Yoshihiko Kayama

Department of Media and Information, BAIKA Women's University
2-19-5 Shukuno-sho, Ibaraki 567-8578, Osaka, Japan
y_kayama@ieee.org

Extended Abstract

Conway's Game of Life, or simply Life, is the most studied two-dimensional binary cellular automaton (Gardner, 1970; Berlekamp et al., 1982). Life has plentiful complex behaviors in class IV of Wolfram's classification of cellular automata (Wolfram, 1983) and has been investigated and applied in many fields. Life also has the property of self-organized criticality (SOC) (Bak et al., 1989). SOC was presented in a general theory of self-organization and emergence by Bak et al. (1987). In many real-world networks of complex systems, Barabási and Albert (1999) subsequently found the property of power-law scaling called scale-freeness. Preferential attachment is a general mechanism for the development of such scale-free networks. Scaling can be recognized as a critical feature of complex self-organized systems.

In our previous articles, we introduced a network representation of cellular automata (CA) that focuses on the effective relationships between cells rather than their states (Kayama, 2010, 2011). Its application to Life showed that the networks derived from typical patterns have a distinctive feature resulting from the dynamical aspect of each pattern (Kayama and Imamura, 2011; Kayama, 2012). In particular, these networks can be divided into two types based on whether or not growth continues. The growing in-links play a crucial role in the network derived for a rest state, which is a critical state evolved from an initial state through a transient self-organization process. The in-links combine residual patterns in the rest state and represent underlying tension that causes avalanches catalyzed by tiny perturbations. Each avalanche can be interpreted as a branch graph of the rest-state network.

In this article, we investigate the dynamical aspects of Life that have been elucidated by its network representation. The well-known patterns are classified by the different aspects of their in-links mentioned above, and the cause of the growing in-links is clarified. While the in-links have an essential role in long-range interactions, out-links illustrate a local area in which the pattern may be affected by the presence of other active cells. The out-links form multi-patterns into a large pattern, which will help organize a hierarchical structure of Life patterns. Moreover, the network of a rest state is investigated to confirm SOC in Life. The size of an avalanche is defined by total changes in out-degrees over its lifetime. The resultant distributions of the lifetimes and sizes of avalanches are in good agreement with those reported previously (Bak et al., 1989; Alström and Leão, 1994). We can show that the network of a rest state has a scale-free nature in both in- and out-degree distributions (Fig. 1). The mechanism for the development of the scale-free nature of the in-links can be explained as the preferential attachment of the surviving cells in the rest state.

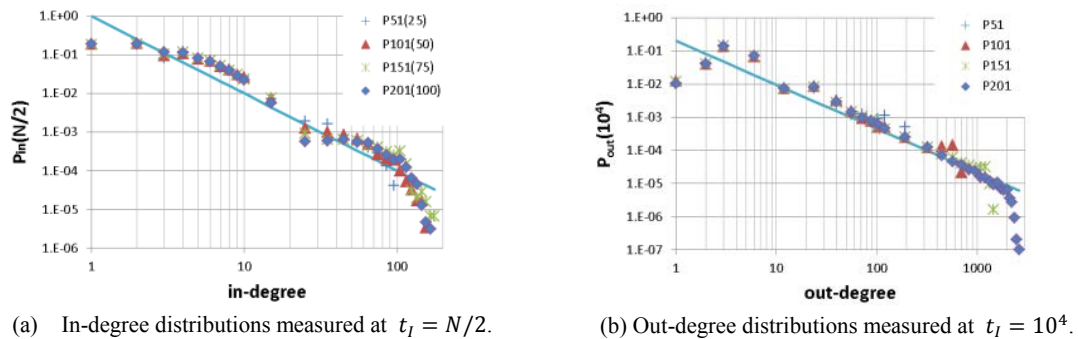


Figure 1: Normalized (a) in-degree distributions and (b) out-degree distributions measured for lattice size $N = 51, 101, 151$, and 201 .

References

- Alström, P. and Leão, J. (1994). Self-organized criticality in the "game of life". *Phys. Rev. E*, 49:R2507–R2508.
- Bak, P., Chen, K., and Creutz, M. (1989). Self-organized criticality in the 'game of life'. *Nature (London)*, 342:780.
- Bak, P., Tang, C., and Wiesenfeld, K. (1987). Self-organized criticality: an explanation of $1/f$ noise. *Physical Review Letters*, 59 (4):381–384.
- Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286:509–512.
- Berlekamp, E. R., Conway, J. H., and Guy, R. K. (1982). *Winning Ways for Your Mathematical Plays*. Academic, New York.
- Gardner, M. (1970). Mathematical games. *Scientific American*, 223:102–123.
- Kayama, Y. (2010). Complex networks derived from cellular automata. arXiv:1009.4509.
- Kayama, Y. (2011). Network representation of cellular automata. In *2011 IEEE Symposium on Artificial Life (IEEE ALIFE 2011) at SSCI 2011*, pages 194–202.
- Kayama, Y. (2012). Network View of Conway's Game of Life. <http://demonstrations.wolfram.com/NetworkViewOfConwaysGameOfLife/> Wolfram Demonstrations Project. Published: May 9, 2012.
- Kayama, Y. and Imamura, Y. (2011). Network representation of the game of life. *Journal of Artificial Intelligence and Soft Computing Research*, 1 (3):233–240.
- Wolfram, S. (1983). Statistical mechanics of cellular automata. *Rev. Mod. Phys.*, 55:601–644.

Continuous *in vitro* Evolution of a Ribozyme Ligase: A Model Kit for The Evolution of a Biomolecule

Michael P. Ledbetter, Tony W. Hwang, Gwendolyn M. Stovall and Andrew D. Ellington

Institute for Cell and Molecular Biology, University of Texas at Austin
Department of Chemistry and Biochemistry, University of Texas at Austin
MichaelLedbetter@utexas.edu, Andy.Ellington@mail.utexas.edu

Extended Abstract

Direct laboratory experience has been found to help in the attainment of many science education goals put forth by the National Research Council.¹ However there is a lack of hands-on demonstrations of Darwinian evolution in progress at both high school and undergraduate level biology courses. This is likely due to the large time frames needed to observe the changes caused by the forces of evolution. In order to address this concern, continuous *in vitro* evolution of a b1-207 ribozyme ligase based pool will be paired with a quadruplex deoxyribozyme peroxidase/ABTS strand displacement reporter system in order to visualize the evolution of improved catalytic function.^{2,3} Ribozyme pools will be taken through rounds of isothermal based amplification dependent on the self-ligation of a T7 promoter.³ Dilution between rounds of evolution will select for ribozymes with faster ligation kinetics.³ As the pool evolves the deoxyribozyme strand displacement system will allow for the monitoring of the pool's ligation rate. The deoxyribozyme strand displacement reporter system allows for visual detection of ligated ribozyme.³ When ligated with the T7 promoter, the 5' end of the ribozyme possesses enough complementarity with the deoxyribozyme to displace, through toehold mediated strand displacement, an inhibitor sequence and stabilize the active conformation of the quadruplex of the deoxyribozyme, which catalyses the conversion of colorless ABTS to green ABTS** in turn producing a visible color change in solution.³ As the ligation rate of the pool increases due to the selection for faster ligating species, the student will observe more rapid development of the green color change in later rounds of evolution. The pairing of the continuous isothermal system with the deoxyribozyme strand displacement detection scheme allows any user, provided with the right starting materials to model the continuous evolution of a biomolecule. The student will simultaneously learn the fundamental principles of Darwinian evolution and be introduced to leading origin of life theories, namely the RNA world, by observing the evolution of superior catalytic function, and in turn reproductive/self-replicative capabilities, at the population level, in our case the pool of ribozymes, under the selective pressure that is applied by the student.⁴ The ribozyme for selection has already been constructed and its viability through rounds of evolution has been confirmed. Initial experiments, with fluorophore and quencher labeled reporter oligos in place of the deoxyribozyme, indicate that the strand displacement system is feasible and specific for detection of ligated ribozyme. Currently, amplification schemes are being tested to increase the detection limits of the scheme to the level of ribozyme concentration found in the amplification reaction.

This material is based in part upon work supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

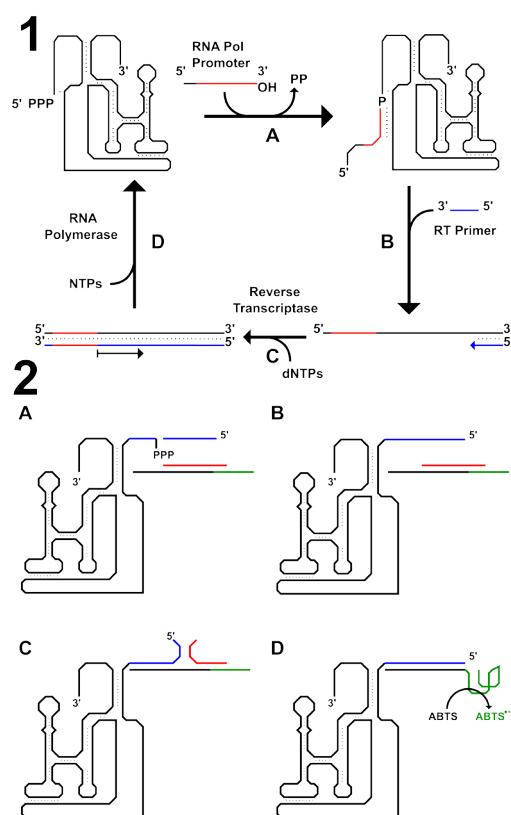


Figure 1: 1 outlines the isothermal continuous evolution scheme of the self ligating ribozymes described above. 2 outlines the deoxyribozyme visualization scheme described above. Blue indicates the region of the ribozyme and its substrate that is complementary to the deoxyribozyme sequence. Red indicates the inhibitor sequence of the deoxyribozyme. Green indicates the catalytic quadruplex region of the ribozyme that is stabilized by the ribozyme.

References

1. Singer, S.R., Hilton, M.L. and Schweingruber, H.A. editors (2006). *America's Lab Report: Investigations in High School Science*. The National Academies Press, Washington, D.C.
2. Wright, M.C. and Joyce, G. F. (1997). Continuous in vitro evolution of catalytic function. *Science*, 276: 614-617.
3. Li, B., Ellington, A. D., Chen, X. (2011). Rational, modular adaptation of enzyme-free DNA circuits to multiple detection methods. *Nucleic Acids Research*. 1-13.
4. Ellington, A. D., Chen, X., Robertson, M. and Syrett, A. (2009). Evolutionary origins and directed evolution of RNA. *The International Journal of Biochemistry and Cell Biology*. 41:254-265.

Differences in the concept of fitness between artificial evolution and natural selection

Paweł Lichocki¹, Laurent Keller² and Dario Floreano¹

¹Laboratory of Intelligent Systems (LIS), Ecole Polytechnique Fédérale de Lausanne (EPFL), Station 11, 1015, Lausanne, Switzerland

²Department of Ecology and Evolution (DEE), University of Lausanne (UNIL), Dorigny, 1015, Lausanne, Switzerland
pawel.lichocki@epfl.ch

Extended Abstract

Evolutionary algorithms were proposed to automatically find solutions to computational problems, much like evolution discovers new adaptive traits (Fogel et al., 1966). Lately, they have been used to address challenging questions about the evolution of modularity (Kashtan et al., 2007), the genetic code (Vetsigian et al., 2006), communication (Floreano et al., 2007), division of labor (Lichocki et al., 2012) and cooperation (Riolo et al., 2001; Waibel et al., 2011). Evolutionary algorithms are increasingly popular in biological studies, because they give precise control over the experimental conditions (Floreano and Keller, 2010) and allow the study of evolution at unprecedented level of detail (Adami, 2006). Nevertheless, evolutionary algorithms have their own caveats, which are often overlooked. Here, we highlight one of them by exposing a terminological conflict between definitions of fitness used in biology and in evolutionary algorithms.

Fitness is a core concept in evolutionary biology (Wagner, 2010). Although used to mean subtly different things (Orr, 2009), it is commonly agreed that fitness is a variable that describes competitive abilities of a given genotype against others in a population under some environmental conditions (Wagner, 2010). The understanding of fitness is very well captured in selection equations (Fisher, 1930; Wright, 1969), where the relative fitness, i.e., the ratio between a fitness value and the mean fitness in a population, directly translates into a proportionate reproductive success. Consequently, only relative fitness bears meaning, i.e., all fitness values may be scaled by the same constant and the evolutionary dynamics would remain the same (Wagner, 2010). For convenience, fitness is usually taken to be the expected or realized number of offspring (Rice, 2004; Orr, 2009).

In contrast to biology, in evolutionary algorithms the term fitness does not usually refer to the reproductive success. Instead, fitness means the performance of a given genotype in solving a given problem. For example, if a genotype encodes a control system that guides a robot's movement in a labyrinth, its performance could be measured as the time needed to find the exit. Once all genotypes are evaluated, they are selected according to their performance val-

ues, and then copied and varied. Several popular selection methods exist: proportionate selection (Goldberg (1989); or roulette wheel selection; used by Waibel et al. (2011)), truncation-proportionate selection (used by Lichocki et al. (2012)), truncation selection (Schlierkamp-Voosen (1993); or (μ, λ) -selection (Back, 1994); used by Floreano et al. (2007); Kashtan et al. (2007)), rank selection (Baker, 1985) and tournament selection (Goldberg and Deb (1991); used by Riolo et al. (2001)).

Here, we experimentally and formally show that the reproductive success of genotypes is proportional to the performance only with proportionate selection. Consequently, only then a genotype's performance, called fitness by evolutionary algorithms practitioners, is actually fitness in the biological sense. All other selection methods introduce a non-linear transformation of performance values into reproductive success. Thus, in all these cases performance is not fitness in the biological sense. This observation has a limited practical meaning in engineering application, where the goal is to find optimal solution to a problem. Usually, the best suited selection method is used and terminological issues are not of any relevance.

In contrast, in biological studies that rely on evolutionary simulations a clear distinction between performance and fitness is necessary for a meaningful interpretation. We support this claim with numerical experiments in which we conducted 1000 generations of artificial selection in groups of agents. Each agent displayed selfish or altruistic behavior towards its teammate. We show that the outcome of the evolutionary simulations of cooperation (i.e., emergence of reproductive division of labor) depends on the selection method and its parameters (Fig. 1).

We considered the evolution of cooperation as our model system, because evolutionary algorithms are a popular tool in this domain (see, e.g., Riolo et al. (2001); Floreano et al. (2007); Waibel et al. (2011)). In the evolution of cooperation, the crucial concepts are cost and benefit of a cooperative act. Importantly, these cost and benefit of cooperation are additive to fitness. In contrast, an experimenter who uses evolutionary simulations may influence costs and benefits

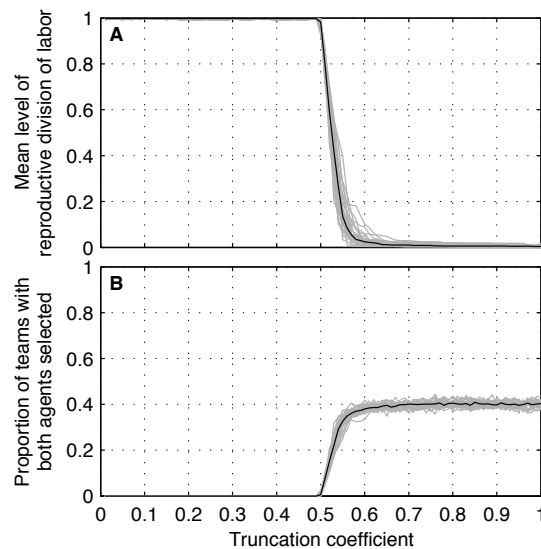


Figure 1: (A) Mean level of reproductive division of labor in population of 500 teams, each consisting of two agents. (B) Proportion of teams in the generation 999 that contributed both agents to the last 1000th generation. Each agent was assigned one performance point by default, and then could transfer it to its partner in the team. The team displayed reproductive division of labor when one agent was selfish, i.e., kept the performance point to itself, and the other agent was altruistic, i.e., gave its performance point to the partner. The evolutionary simulation was replicated 30 times for each of the 100 treatments (truncation coefficient was set to a value from 0.01 to 1, with a step of 0.01). The result of each replicate is shown in grey.

additive to performance. Consequently, in order to validate the predictions of biological models of cooperation, a correction for the selection method must be applied to fitness, in the case of a non-proportionate selection. Alternatively, one may use proportionate selection. Then, performance is fitness, and cost and benefit additive to performance are automatically additive to fitness. Note, however, that proportionate selection is known to display several disadvantageous properties, e.g., premature convergence (Baker, 1987).

Overall, we call for caution when using evolutionary algorithms in biological studies and advise to carefully account for effects that a selection method has on the fitness landscape.

Acknowledgments

This work was supported by the Swiss National Science Foundation (grant number 200021_127143)

References

- Adami, C. (2006). Digital genetics: unravelling the genetic basis of evolution. *Nature Reviews Genetics*, 7(2):109–118.
- Back, T. (1994). Selective pressure in evolutionary algorithms: A

characterization of selection mechanisms. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pages 57–62. IEEE.

Baker, J. (1985). Adaptive selection methods for genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 101–111. L. Erlbaum Associates Inc.

Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In *Proc. of the 2nd Intl Conf on GA*, pages 14–21. Lawrence Erlbaum Associates, Inc. Mahwah, NJ, USA.

Fisher, R. (1930). *The genetical theory of natural selection*. Clarendon Press, Oxford, England.

Floreano, D. and Keller, L. (2010). Evolution of adaptive behaviour in robots by means of darwinian selection. *PLoS biology*, 8(1):e1000292.

Floreano, D., Mitri, S., Magnenat, S., and Keller, L. (2007). Evolutionary conditions for the emergence of communication in robots. *Current biology*, 17(6):514–519.

Fogel, L. J., Owens, A. J., and Walsh, M. J. (1966). *Artificial Intelligence through Simulated Evolution*. John Wiley and Sons, New York.

Goldberg, D. and Deb, K. (1991). *A comparative analysis of selection schemes used in genetic algorithms*, pages 69–93. Morgan Kaufmann Publishers, San Mateo, CA, USA.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Addison Wesley, Boston, MA, USA.

Kashtan, N., Noor, E., and Alon, U. (2007). Varying environments can speed up evolution. *Proceedings of the National Academy of Sciences*, 104(34):13711.

Lichocki, P., Tarapore, D., Keller, L., and Floreano, D. (2012). Neural networks as mechanisms to regulate division of labor. *The American Naturalist*, 179(3):391–400.

Orr, H. (2009). Fitness and its role in evolutionary genetics. *Nature Reviews Genetics*, 10(8):531–539.

Rice, S. (2004). *Evolutionary theory: mathematical and conceptual foundations*. Sinauer Associates, Sunderland, MA, USA.

Riolo, R., Cohen, M., and Axelrod, R. (2001). Evolution of cooperation without reciprocity. *Nature*, 414(6862):441–443.

Schlierkamp-Voosen, D. (1993). Predictive models for the breeder genetic algorithm. *Evolutionary Computation*, 1(1):25–49.

Vetsigian, K., Woese, C., and Goldenfeld, N. (2006). Collective evolution and the genetic code. *Proceedings of the National Academy of Sciences*, 103(28):10696.

Wagner, G. (2010). The measurement theory of fitness. *Evolution*, 64(5):1358–1376.

Waibel, M., Floreano, D., and Keller, L. (2011). A quantitative test of hamilton's rule for the evolution of altruism. *PLoS Biology*, 9(5):e1000615.

Wright, S. (1969). *Evolution and the genetics of populations. Volume 2: the theory of gene frequencies*. The University of Chicago Press, Chicago, IL, USA.

Key innovation in a virus catalyzes a coevolutionary arms race

Justin R. Meyer^{1*}, Cesar Flores², Joshua S. Weitz^{2,3}, Richard E. Lenski¹

¹BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, MI

²School of Physics, Georgia Institute of Technology, Atlanta, GA

³School of Biology, Georgia Institute of Technology, Atlanta, GA

*justin.raymond.meyer@gmail.com

Extended Abstract

I believe that ... any replay of the tape would lead evolution down a pathway radically different from the road actually taken. – Stephen J. Gould¹

Starting with identical organisms under identical conditions, replicate evolution experiments nonetheless sometimes produce very different outcomes^{2,3}. These differences arise because mutations occur randomly, such that variation in their timing and order of appearance can affect evolutionary trajectories. An example of contingency was reported to occur when a virus, phage λ , coevolved with its host, the bacterium *Escherichia coli*³. Meyer *et al.* found that λ evolved the ability to infect *E. coli* cells through a new receptor, the host's OmpF protein, in about one-quarter of the replicate communities, but it did not do so in the others. This bifurcation in the virus's evolution occurred, at least in part, because of variation in the mutations that evolved in the host populations. Here we compare the patterns of coevolution in two communities including one where the virus evolved this key innovation and one where it did not. In the community where the virus evolved the innovation a coevolutionary arms race immediately ensued, whereas the other community appeared static. This work illustrates how early variation in the sequence of evolutionary events can propagate to produce increasingly different outcomes.

When communities of *E. coli* and λ were cultured in a glucose-limited medium, the bacteria evolved resistance through mutations in *malT* within about a week. The *malT* mutations caused reduced expression of the protein LamB, which the phage uses to attach to and inject its DNA through the host's outer cell envelope⁴. Over time, λ then evolved mutations in its J gene, which encodes the protein it uses to bind to LamB, and these mutations evidently improved its ability to infect the host cells⁴. At this stage, the evolutionary dynamics diverged quite sharply. Some phage populations evolved a combination of four J mutations that conferred the ability to use OmpF as an alternative receptor. However, most of the bacterial populations evolved a mutation in *manXYZ*, a set of genes encoding a channel that the phage uses to transport its DNA across the host's inner membrane. If one of these mutations reached high frequency in the host population, then the phage population did not evolve the ability to target OmpF, thereby leaving it dependent on the ancestral receptor. This divergence typically occurred within about two weeks, after which λ and *E. coli* continued to coexist for at least two more weeks.³

In the present study, we examined the coevolutionary dynamics associated with the phage's innovation or the lack thereof. To do so, we revived the ancestral *E. coli* and λ along with samples from two communities that had been frozen on days 8, 15, 22 and 28 of the initial experiment reported by Meyer *et al.* Phage λ evolved the ability to exploit OmpF in one of the communities (Fig. 1A), but not in the other (Fig. 1B). We isolated 10 phage and 10 bacterial clones from each time-point and community. We then challenged each bacterial clone with each phage isolated from the same replicate community. We measured the ability of each phage to infect an evolved bacterium relative to its ability to infect the ancestral bacterium; this infectivity metric has also been called the efficiency of plaquing⁵. Figure 1 shows these data as interaction matrices. Figure 2 summarizes how the average bacterial resistance and phage host-range changed over time in the community where the phage evolved the ability to use OmpF, including both contemporaneous and time-shifted interactions.

The differences between the two communities are striking. The virus evolved the ability to use the new OmpF receptor at about day 8, which led to the emergence of a diverse assemblage of bacteria and phage with different patterns of resistance and infectivity, respectively (Fig. 1A). Over time, the bacteria evolved increasing resistance to the phage (Fig. 2), including two clones from day 28 that were completely resistant to all 40 phage isolates. Also, the phage tended to evolve expanded host ranges (Fig. 2) and increased infectivity on the bacteria, with these trends being more pronounced when phage from later generations were tested on bacteria from earlier generations.

Key innovations are not end-points in evolution, but instead they are hypothesized to spark further rapid change. For example, the processes of speciation and adaptive radiation are thought often to follow the evolution of new ecological functions⁶. We have shown here that a key innovation in an evolving virus population catalyzed a coevolutionary arms race with its host, which led to the rapid diversification of both the host and parasite. Although the two communities that we studied began with

identical ancestors and experienced identical abiotic environments, they followed radically different evolutionary trajectories, consistent with Gould’s view of life¹.

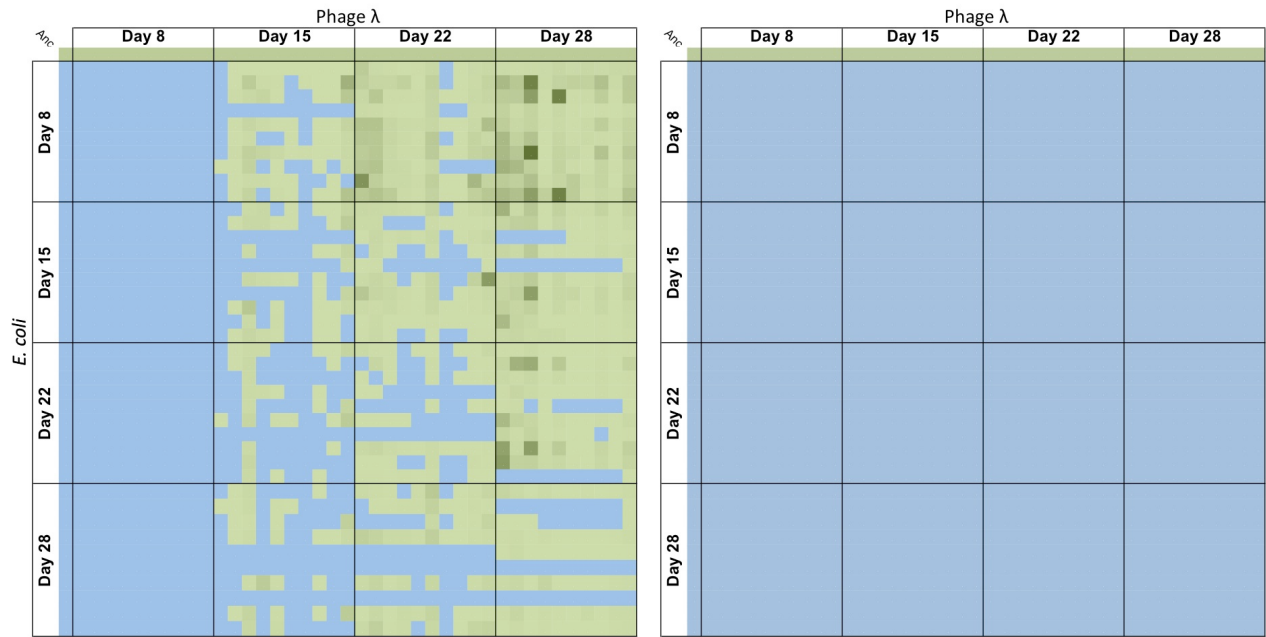


Fig. 1. Interaction matrices for phage λ and *E. coli* from two replicate communities. (A) Community in which λ evolved the new ability to use OmpF as a receptor. **(B)** Community in which λ remained dependent on the ancestral receptor, LambB. In each panel, 41 bacterial clones (the ancestor and 10 from each of four later time points) are arranged vertically and 41 phage isolates (the ancestor and 10 each from the same time points) are arranged horizontally. Each cell shows the estimated ability of the phage isolate to infect the bacterial clone; blue indicates no observable plaques, light green indicates minimal plaque formation, and darker shades of green indicate greater infectivity based on plaque formation relative to the same phage isolate’s performance on the ancestral host.

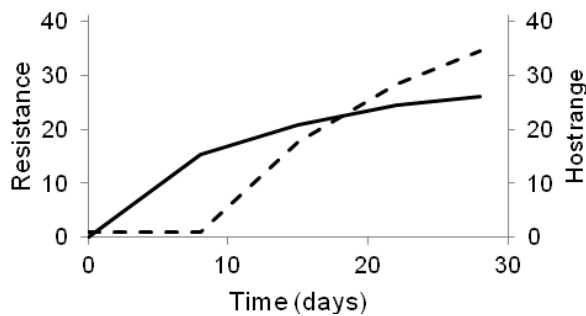


Fig. 2. Summary statistics for bacterial resistance and phage host range for the community shown in Figure 1A. The solid line shows the average resistance of the bacterial clones over time based on the 10 clones at each time point (except day 0 for which there is one ancestral clone) and the number of the 41 phage isolates that could not infect that clone at all (blue cells in Fig. 1A). The dashed line shows the average host-range of the phage isolates over time based on the 10 isolates at each time point (except day 0 for which there is one ancestral isolate) and the number of the 41 bacterial clones that the phage isolate could infect (all green cells in Fig. 1A).

References

- 1 Gould, S. J. *Wonderful Life: The Burgess Shale and the Nature of History* (Norton, New York, 1989).
- 2 Blount, Z. D., Borland, C. Z. & Lenski, R. E. Historical contingency and the evolution of a key innovation in an experimental population of *Escherichia coli*. *Proc. Natl. Acad. Sci. U.S.A.* **105**, 7899-7906, doi:10.1073/pnas.0803151105 (2008).
- 3 Meyer, J. R. *et al.* Repeatability and contingency in the evolution of a key innovation in phage Lambda. *Science* **335**, 428-432, doi:10.1126/science.1214449 (2012).
- 4 Katsura, I. *Tail Assembly and Injection* (Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, 1981).
- 5 Adams, M. H. *Bacteriophages* (Interscience Publishers, New York, 1959).
- 6 Schluter, D. *The Ecology of Adaptive Radiation*. (Oxford University Press, New York, 2000).

The Evolution of Allosteric Cooperativity in a Simple Artificial Life System

A.M. Novak¹, A.E. Clark¹, C.M. DeBoever¹, L.E. Haynes¹, S. Ma¹, M. McDermott¹, J.S. Wentworth¹ and E.C Bush¹

¹Biology Department, Harvey Mudd College, Claremont CA 91711
bush@hmc.edu

Extended Abstract

Most enzymes operate in multi-subunit complexes. Many, perhaps 25% or more, display allosteric cooperativity in binding ligand molecules (Hopkinson et al., 1976; Hill et al., 1977; Traut and Evans, 1988; Traut, 1994). In this situation, the binding of one subunit to its ligand can influence the affinity of other subunits, altering the shape of the ligand binding curve (fig. 1 A).

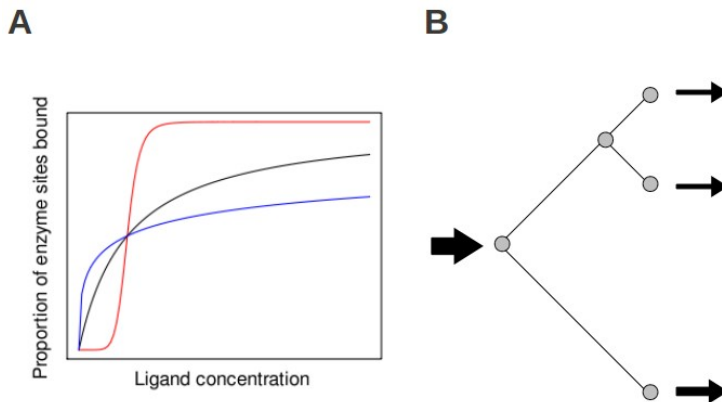


Figure 1: A. Ligand binding curves for no cooperativity (black), positive cooperativity (red), and negative cooperativity (blue). B. A simple metabolic network with metabolites (circles) and reactions (lines). Arrows indicate the locations of inflow and outflow to the network.

Such effects undoubtedly have an important impact on metabolic regulation. This is easiest to understand in the case of positive cooperativity. Sharp responses to changes in metabolite concentrations allow organisms to better respond to environmental changes and maintain metabolic homeostasis. However, despite the fact that negative cooperativity is almost as common as positive, it has been harder to imagine what advantages it provides (Koshland and Hamadani, 2002). One hypothesis suggests it is associated with branch points in metabolic networks (Koshland, 1996).

We have developed an artificial life system to examine this hypothesis in the context of enzyme-inhibitor binding. Our system operates in simple branching metabolic networks (fig. 1 B). These are subject to environmental variation in the form of differing rates of inflow and outflow. An organism consists of a set of enzymes whose goal is to maintain metabolic homeostasis. Here we have used a population size of 100. The initial population has randomly chosen inhibitor binding characteristics. In each generation the top 30 organisms are selected to reproduce. We carry the simulation forward 500 generations, in which time fitness improves substantially.

We carried out 264 simulations on the network in fig. 1 B. From each we obtained the most fit organism in the final generation. Among these, cooperativity at branch and non-branch enzymes differs significantly (fig. 2 A & B). We model inhibitor binding with the Hill equation, and in our experiments, hill coefficients can range from 0.1 to 10. At non-branch enzymes median hill values from our 264 most fit organisms were 10 corresponding to strong positive cooperativity. For these enzymes, the sharper the response the better. In contrast at the branch point enzyme median hill coefficients were 7.54 and 7.47 for the two inhibitors. Thus branch point enzymes have reduced levels of positive cooperativity compared to non-branch enzymes. This phenomenon likely results from the need to integrate signals from multiple metabolites.

We have not yet demonstrated negative cooperativity (hill values between 0 and 1). However the factors which cause reduced positive cooperativity in our system could easily contribute to negative cooperativity in other circumstances. A goal for the future is to explore this possibility.

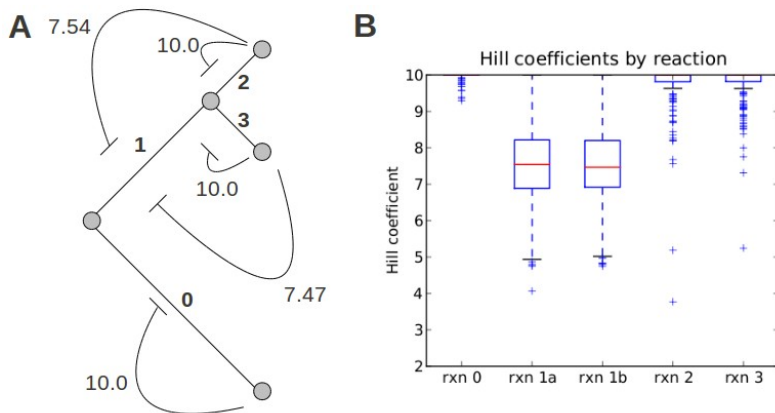


Figure 2: A. Metabolic network with reactions numbered and inhibitory relationships indicated. Inhibitors are labeled with the median hill coefficient from our set of 264 most fit organisms. B. Boxplots of hill coefficients from the most fit organisms, reactions numbered as in A.

References

- Hill, C.M., Waighm, R.D., and Bardsley, W.G. (1977). Does any enzyme follow the Michaelis-Menten equation? *Molecular and Cellular Biochemistry*, 15(3):173–178.
- Hopkinson, D.A., Edwards, Y.H., and Harris, H. (1976). The distributions of subunit numbers and subunit sizes of enzymes: a study of the products of 100 human gene loci. *Annals of human genetics*, 39(4):383–392.
- Koshland, D. E. and Hamadani, K. (2002). Proteomics and models for enzyme cooperativity. *Journal of Biological Chemistry*, 277(49):46841–46844.
- Koshland, D. E. (1996) The structural basis of negative cooperativity: receptors and enzymes. *Current Opinion in Structural Biology*, 6(6):757–761.
- Traut, T.W. (1994). Dissociation of enzyme oligomers: a mechanism for allosteric regulation. *Critical Reviews in Biochemistry and Molecular Biology*, 29(2):125–163.
- Traut, T.W. and Evans, D.R. (1988) . Enzymes of nucleotide metabolism: the significance of subunit size and polymer size for biological function and regulatory properties. *Critical Reviews in Biochemistry and Molecular Biology*, 23(2):121–169.

The Paradoxical Effects of Allelic Recombination on Fitness

David P. Parsons^{1,3}, Carole Knibbe^{2,3} and Guillaume Beslon^{1,3}

¹Université de Lyon, CNRS, INRIA, INSA-Lyon, LIRIS, UMR5205, F-69621, France

²Université de Lyon, CNRS, INRIA, Université Lyon 1, LIRIS, UMR5205, F-69622, France

³IXXI, Institut Rhône-Alpin des Systèmes Complexes, Lyon, F-69007, France
guillaume.beslon@liris.cnrs.fr

Extended Abstract

Introduction

Horizontal transfer (HT) plays a major role in bacterial evolution, providing a way for bacteria to take advantage of beneficial mutations found by other bacteria, possibly from other species. Within a given species, horizontal transfers allow bacteria to evade the clonal interference phenomenon (Hill and Robertson, 1966) through allelic recombination: when two different beneficial mutations are found concomitantly in two different lineages, horizontal transfer allows both mutations to be assembled into a single organism, thus speeding up evolution. Transfer also enables the isolation of the “ruby in the rubbish” (Peck, 1994): beneficial mutations being very rare compared to deleterious ones, it is likely that deleterious mutations will happen at the same time as a beneficial one, thus overwhelming the benefits of the latter. Transfer however, allows to solve this problem by breaking the linkage between the affected alleles.

In this work, focusing on transfer involving recombination rather than simple plasmid exchange, we used the Aevol model to study the influence of HT on the evolution of both fitness and genomic architecture. The Aevol model is a digital genetics model which is realistic at the level of the genome but abstract at the phenotypic level: each individual has a double stranded genome upon which genes are detected through signal sequences and a transcription-translation process. These genes are then interpreted in a mathematical formalism and combined to solve a curve-fitting task (Knibbe et al., 2007).

Experiments

We let 105 populations of 1,000 individuals evolve independently for 50,000 generations with the same curve-fitting task. Each population was seeded with a random binary sequence of 5,000 bp containing at least one “good” gene. At each replication, the genome could undergo point mutations, indels (up to 6 bp) and chromosomal rearrangements (duplications, deletions, translocations and inversions) with random breakpoints (7 rates tested, from 10^{-6} to 10^{-4} per base). In addition, we tested 3 different schemes of HT, thus forming 3 groups of simulations. In group A, at each

replication, a transfer attempt was conducted with probability 0.1. A transfer attempt consists in trying to replace a sequence of the form $(end1)(anysequence)(end2)$ in the (replicating) recipient genome by a sequence with similar ends $(\sim end1)(anysequence)(\sim end2)$ from the (randomly chosen) donor genome. Note that because the regions that need to be similar are limited to the sequences around the breakpoints and not the whole sequence, the transferred and the replaced sequences may differ greatly in length and content. A simple match/mismatch scoring function (no gaps) was used: highly similar sequences ($score > 30$) were given a high probability of leading to a transfer event (homologous recombination) while regions of low similarity were only assigned a low, although not null, probability (nonhomologous recombination). This model of HT is similar to the homology driven chromosomal rearrangement model described in (Parsons et al., 2011). In the second group of simulations (HT scheme B), transfers were deterministically triggered between random points at the same rate as that effectively observed in group A. Finally, in group C, transfer was completely disabled.

Results

We analysed the transfer events that occurred during the whole evolution and found that the sensitivity to sequence similarity proves to favour those transfers whose involved segments (transferred and replaced segments) are of roughly the same size (figure 2). It appears that many transfers consist in replacing a given sequence by another sequence of exactly the same size. We also observe that there are more transfers involving sequences that differ by only one to six bases in length than there are with greater differences. This is of particular interest since in these experiments, the maximum size of an indel is of precisely six. This strongly suggests that both sequences are homologous, having undergone only point mutations and at most one indel. It hence appears that alignment driven transfer does indeed promote allelic recombination.

The distribution of the scores of the alignments that lead to either beneficial, neutral or deleterious transfers in group

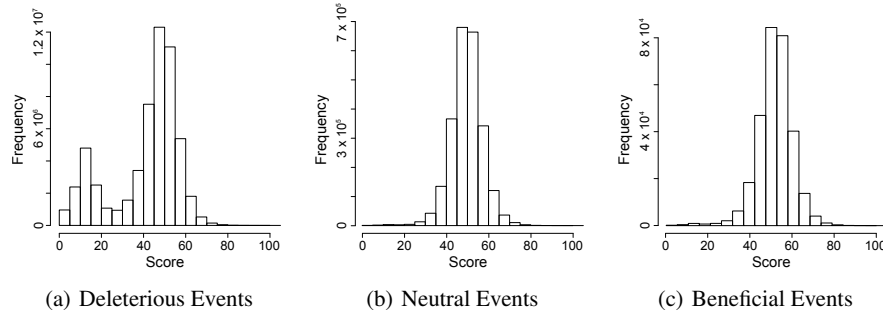


Figure 1: Distribution of the score of the alignments that lead to a **(a)**: deleterious, **(b)**: neutral and **(c)**: beneficial transfer.

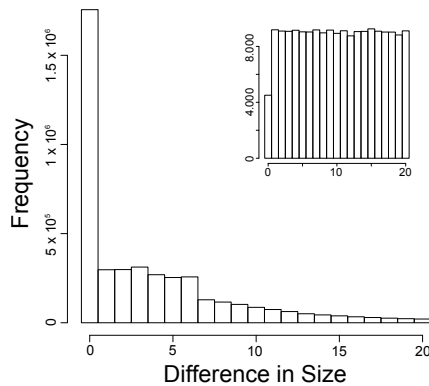


Figure 2: Distribution of the difference in size between the transferred and the replaced sequence for alignment driven transfer (group A). Inset: distribution for random point transfer (group B).

A (figure 1) is of great interest: almost all the replications involving transfer that either improved the fitness or were neutral correspond to the exchange of segments with highly similar ends ($score > 30$) while most of the exchanges with weakly similar ends had deleterious effects. As a matter of fact, the proportion of both neutral and beneficial replications among those involving transfer was higher by up to two orders of magnitude in the case of homology driven transfer (group A) than in the case of random point transfer (group B – data not shown).

Surprisingly, even though homology driven transfer has proved to allow for allelic recombination, and despite all the theoretical benefits it could confer, there seems to be very little (if any) differences in the fitness of the evolved organisms between the different groups of simulations. We conducted a statistical analysis (multiple linear regression with Student's t-tests on the coefficients, Kruskal-Wallis test) of the fitnesses of the final best organism of each population. These tests show that the HT scheme has no significant effect on fitness after 50,000 generations. Actually, the only parameter that significantly affects fitness is the rearrangement rate, which supports our previous results (Knibbe et al., 2007) on the impact of rearrangement rates on evolution.

This lack of effect of transfer on the outcome of evolution in terms of fitness comes as a paradox when considered in the light of the apparent benefit of allelic transfer at the individual level. Indeed, it could be expected that group A would benefit from transfer since it was shown to allow for fitness improvements. The fact that this fails to happen could be explained by different hypotheses: the coalescence time in these experiments seems to be very short, which suggests a regime of successive rather than parallel mutations. This means that clonal interference might be very rare in these experiments. Also, even though transfer is beneficial more frequently when alignments are involved, it remains mostly deleterious. Given that in our experiments, transfers are rare, it is clear that beneficial transfers are very rare and might not make any difference in the long term.

Future experiments will thus aim at assessing under which conditions transfer can be beneficial on the population level.

Acknowledgements

We gratefully acknowledge support from the CNRS/IN2P3 Computing Center (Lyon/Villeurbanne - France), for providing most of the computing resources needed for this work. We also thank the CNRS-PEPS and CNRS-PEPII programs.

References

- Hill, W. G. and Robertson, A. (1966). The effect of linkage on limits to artificial selection. *Genetics Research*, 8(03):269–294.
- Knibbe, C., Coulon, A., Mazet, O., Fayard, J.-M., and Beslon, G. (2007). A long-term evolutionary pressure on the amount of noncoding DNA. *Mol. Biol. Evol.*, 24(10):2344–2353.
- Parsons, D. P., Knibbe, C., and Beslon, G. (2011). Homologous and nonhomologous rearrangements: Interactions and effects on evolvability. In *Proceedings of ECAL 11*, pages 622–629.
- Peck, J. R. (1994). A ruby in the rubbish: beneficial mutations, deleterious mutations and the evolution of sex. *Genetics*, 137(2):597–606.
- Sniegowski, P., Gerrish, P., Johnson, T., and Shaver, A. (2000). The evolution of mutation rates: Separating causes from consequences. *Bioessays*, 22:1057–1066.

Quantifying Frequency-Dependent Fitness Effects in Evolving Microbial Populations

Noah Ribeck^{1,2} and Richard E. Lenski^{1,2}

¹Department of Microbiology and Molecular Genetics, Michigan State University, East Lansing, MI 48824, USA

²BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, MI 48824, USA
 ribeck@msu.edu

Extended Abstract

Experimental studies of evolution have shown that microbial populations often generate and sustain stable polymorphisms (Helling et al. 1987; Turner et al. 1996; Elena and Lenski 1997; Treves et al. 1998; Rozen and Lenski 2000; Blount et al. 2008). Indeed, it has been suggested that maintenance of genetic diversity by cross-feeding phenotypes is robust and widespread in evolving bacterial populations (Pfeiffer and Bonhoeffer 2004; Estrela and Gudelj 2010). However, measuring the relative fitness of individual genotypes or clones in such an ecological system presents experimental complications. For microbes, fitness is typically measured by the change in relative frequencies of two populations placed in competition with each other (Lenski 1988; Lenski et al. 1991). However, if fitness is frequency-dependent, then each competitor's fitness changes over the course of the competition, confounding the measurement. To address this problem, we present a theoretical basis for measuring the functional form of the frequency-dependent fitness of two competing types within an asexual population.

We consider a Wright-Fisher type model of allele dynamics in an asexual population containing two genotypes, where the fitness of one type relative to the other decreases monotonically with increasing frequency p . The simplest parameterization of this ecological interaction is a linear, antisymmetric function of relative fitness: the minority type has a selective advantage s near $p = 0$, decreasing linearly until it is selectively neutral at $p = 0.5$, and approaching a disadvantage s at $p = 1$. Ignoring the effect of stochastic fluctuations in a finite population, the two types would eventually settle into a stable coexistence with each occupying half the population, given that other beneficial mutations have not had sufficient time to reach substantial frequency in either clade.

Measuring s in this system requires competing the two clades and measuring the resulting change in relative abundance. Accounting for the changing fitness over the course of the competition, the dynamics of the system can be solved in the small s limit ($s \leq 0.1$). In this regime, which includes all of the examples in the above references, s is given by:

$$1 + s = \left(\frac{g(p_0)}{g(p_n)} \right)^{1/n},$$

where p_0 and p_n are the initial and final frequencies of the minority clade in an n -generation competition, and $g(p)$ is the quantity:

$$g(p) = \frac{(1 - 2p)^2}{(1 - p)p}.$$

This equation can be inverted to express allele frequency as a function of time (Fig. 1). The expression for s can also be generalized to any slope of the frequency-dependent fitness profile, in which case the equilibrium coexistence would not necessarily be at 50%, and at least two measurements at different initial frequencies would be required in order to estimate s and predict the equilibrium.

Further, if the assumption of linearity is not desired, the fitness profile can be represented by a more general form (such as a polynomial) and can be solved by fitting the data to a numerical calculation of relative frequency. This technique is also robust, but it requires a series of competitions at a range of different initial ratios.

We expect this method to be useful for quantitatively characterizing evolving microbial populations. For example, it might be used to test general theories of evolutionary stable states and models of specific mechanisms such as cross-feeding (Doebeli 2002; Bull and Harcombe 2009; Estrela and Gudelj 2010). Finally, our explicit solution of the allele frequency trajectory may enable us to introduce frequency-dependent selection into more general theories of population dynamics.

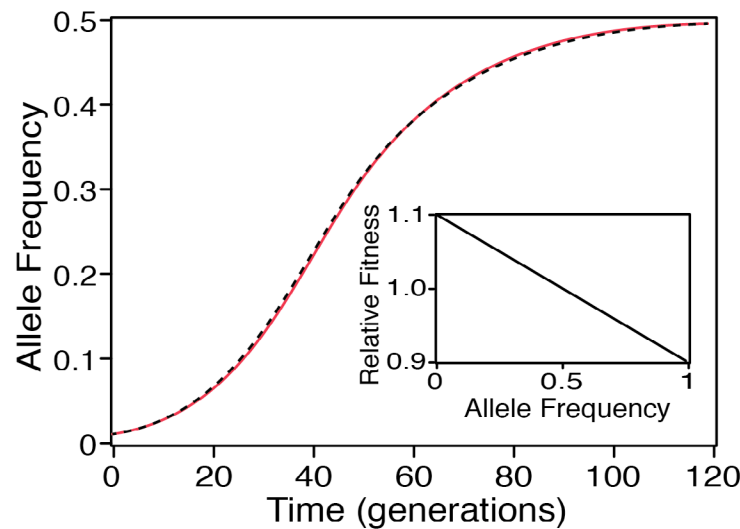


Figure 1. Trajectory of the frequency of the minority clade with $s = 0.1$, starting at $p_0 = 0.01$. The system approaches an equilibrium at $p = 0.5$. The dashed black line is the theoretical solution for small s , and the solid red line is the exact numerical solution. The inset shows the linear, antisymmetric profile of frequency-dependent fitness, also with $s = 0.1$.

References

- Blount, Z. D., C. Z. Borland and R. E. Lenski (2008). Historical contingency and the evolution of a key innovation in an experimental population of *Escherichia coli*. *Proceedings of the National Academy of Sciences of the United States of America*, 105: 7899-7906.
- Bull, J. J. and W. R. Harcombe (2009). Population dynamics constrain the cooperative evolution of cross-feeding. *PLoS ONE*, 4: e4115.
- Doebeli, M. (2002). A model for the evolutionary dynamics of cross-feeding polymorphisms in microorganisms. *Population Ecology*, 44: 59-70.
- Elena, S. F. and R. E. Lenski (1997). Long-term experimental evolution in *Escherichia coli*. VII. Mechanisms maintaining genetic variability within populations. *Evolution*, 51: 1058-1067.
- Estrela, S. and I. Gudelj (2010). Evolution of cooperative cross-feeding could be less challenging than originally thought. *PLoS ONE*, 5: e14121.
- Helling, R. B., C. N. Vargas and J. Adams (1987). Evolution of *Escherichia coli* during growth in a constant environment. *Genetics*, 116: 349-358.
- Lenski, R. E. (1988). Experimental studies of pleiotropy and epistasis in *Escherichia coli*. I. Variation in competitive fitness among mutants resistant to virus T4. *Evolution*, 42: 425-432.
- Lenski, R. E., M. R. Rose, S. C. Simpson and S. C. Tadler (1991). Long-term experimental evolution in *Escherichia coli*. I. Adaptation and divergence during 2,000 generations. *American Naturalist*, 138: 1315-1341.
- Pfeiffer, T. and S. Bonhoeffer (2004). Evolution of cross-feeding in microbial populations. *American Naturalist*, 163: E126-E135.
- Rozen, D. E. and R. E. Lenski (2000). Long-term experimental evolution in *Escherichia coli*. VIII. Dynamics of a balanced polymorphism. *American Naturalist*, 155: 24-35.
- Treves, D. S., S. Manning and J. Adams (1998). Repeated evolution of an acetate-crossfeeding polymorphism in long-term populations of *Escherichia coli*. *Molecular Biology and Evolution*, 15: 789-797.
- Turner, P. E., V. Souza and R. E. Lenski (1996). Tests of ecological mechanisms promoting the stable coexistence of two bacterial genotypes. *Ecology*, 77: 2119-2129.

Exploring the Concept of Open-Ended Evolution

Tim Taylor^{1,2}

¹Department of Computing, Goldsmiths, University of London

²Institute for Informatics and Digital Innovation, Edinburgh Napier University
tim@tim-taylor.com

Extended Abstract

The term *open-ended evolution* (“OEE”) is used by the ALife community to refer to the kind of long-term evolutionary dynamics observed in the biosphere. It is generally taken to refer to evolutionary systems which display a continual production of adaptively significant innovations. Furthermore, some authors use the term to imply a sustained increase in complexity and/or diversity of some components of the evolving system; a system capable of open-ended evolution could spontaneously generate rich ecosystems of complex organisms.

For ALife practitioners who seek to build virtual worlds capable of OEE, there is a need for a *particular type of understanding* of the issues involved; in addition to the *analytic* understanding of evolutionary dynamics provided by theoretical biologists, there is also the need for a *synthetic* understanding of how to design systems that can produce these dynamics. In the following paragraphs, an attempt is made to unpack the concept of OEE into a number of separate (but related) issues, with particular focus on issues which apply to the synthesis of OEE systems.

Basic requirements

A number of common themes are apparent in previous work on OEE. At a very general level, three basic requirements can be identified for an evolutionary system if it is to exhibit the continual appearance of new adaptive forms:

1. A practically unlimited space of potential phenotypes.

Clearly, if a system is to be capable of the continual production of new organisms without practical limit, there should be an unlimited space of potential organisms that could be represented in the medium. It is usually assumed that this requires a mechanism with the potential for transmitting an unlimited amount of genetic information from one generation to the next; that is, unlimited heredity replicators. However, Waddington (1969) and others emphasize the two-way interaction between genetic information and the environment in determining the adult form of an organism. In this case, where the same genotype can produce different phenotypes in different environments,

unlimited heredity may not be strictly necessary if there exists an unlimited variety of potential environments.

2. Mutational pathways of practically unlimited length between potential phenotypes.

It is insufficient to require just an unlimited space of potential phenotypes; these potential forms must be reachable by the evolutionary process. OEE requires that pathways of practically unlimited length exist in this possibility space from the original ancestral organisms to an wide variety of possible future organisms. The shape of the adaptive landscape will depend upon the nature of the information transmitted from parent to offspring, on the properties of the evolutionary operators (e.g. mutation and recombination), on the way in which an adult organism is generated from this information, and on the properties of the environment. These factors will interact in complex ways to determine the properties of the adaptive landscape with respect to features such as neutrality and portals to new adaptive landscapes (Schuster, 2011; Crutchfield, 2003).¹

3. Changing adaptive landscapes to drive continual evolution.

The first two requirements endow a system with the *potential* for OEE. If that potential is to be realized, without external assistance, the system must generate an intrinsic *drive* for continual adaptive evolution. This requires that the adaptive landscape experienced by organisms is changing rather than static, at least over evolutionary time scales. A changing adaptive landscape can come about intrinsically if the fitness of an organism depends on its local environment rather than on the organism in isolation. This can be introduced into a virtual world through the property of *connectedness*, described below.

¹von Neumann (1966) proposed an architecture that theoretically allows mutational pathways of unlimited length, although this kind of architecture would appear to be unnecessary in digital worlds lacking complex environmental dynamics, such as *Tierra* (Ray, 1991), where replication by self-inspection seems to be sufficient.

Connectedness

The fitness of an organism will depend on its local environment if there is a *connectedness* between organism and environment. Such connectedness can come about if organisms engage in the consumption, transformation and excretion of nutrients and energy, creating a food web which connects a whole ecosystem of organisms. Connectedness can also come about by physical aspects of the environment, such as the transmission of forces, the transmission of signals, or modification of physical aspects of an ecosystem. The effect of connectedness, however it is achieved, is that changes in the behavior of one organism in the system, or the introduction of a new type of organism, or removal of an existing type, will have significant consequences for other organisms in the system. Connectedness therefore means that organisms in an ecosystem live in a delicate balance, and evolutionary change in one species will change the adaptive landscape of other species in the ecosystem.

In order to achieve connectedness through the emergence of food webs, the elementary material resources in the system must be conserved. If it is possible to create new resources “out of thin air” (as in *Tierra* when a program writes a new copy of itself in memory), then there is no need for resources to be recycled, and hence no need for food webs; such systems will therefore lack this type of connectedness.

Particular kinds of connectedness can also promote the evolution of diversity and complexity in the system. For example, a predator-prey relationship can lead to an evolutionary increase in complexity of the species involved (Van Valen, 1973). It has also been argued that connectedness through physical ecosystem engineering can result in a net increase in species diversity over long time scales (Jones et al., 1997).

Final comments

In the ways described above, the various forms of connectedness between individuals in an evolving system can lead to changing adaptive landscapes, which drive continual evolution. However, when designing a system that might be capable of OEE, there are additional important considerations to take into account. These are hinted at by the requirements listed above, and include:

1. **A complex physical environment.** OEE can be promoted by providing an environmental medium that can support rich, complex features and processes. This can help OEE in a number of ways, many of which have been discussed above (e.g. by supporting connectedness through food webs, and by providing mechanisms for communication via environment-mediated signals). The richer the range of phenomena available in the environment, the richer the potential for organisms to evolve ways of capturing and manipulating these phenomena for their own purposes. Not only does complexity in the physical environment expand the range of possible organism behaviors, but it also means that the full specification of complex behaviors can be distributed between the organism’s genetic information, and the physics of the environment (thereby reducing the required information capacity of the genome).
2. **Embeddedness of organisms in the environment.** If some parts of the organism are reproduced automatically according to a specific mechanism (i.e. not embedded in the medium of the environment), there must be a predefined procedure to decide *when and how* such a mechanism operates. Such parts will therefore not be subject to variation and evolution, or, at best, only subject to evolve in certain predefined ways. In order to avoid any hard-wired restrictions on evolvability, the organisms must therefore be *fully embedded* in the shared medium of the world. Only then will all aspects of the organism, including its very organization, mode of reproduction, etc., be evolvable. Depending on the design goals of the system, one might choose to forgo total evolvability in the interests of more easily achieving particular outcomes.

Lack of space prevents further elaboration of these issues here; a detailed examination is presented in (Taylor, 2013). The present discussion has at least highlighted that the design of virtual worlds with a capacity for OEE requires much more than the consideration of information processing capacities, including careful consideration of the nature of the relationship between organisms, and of the relationship between an organism and its physical environment.

References

- Crutchfield, J. P. (2003). When evolution is revolution. In Crutchfield, J. P. and Schuster, P., editors, *Evolutionary Dynamics: Exploring the Interplay of Selection, Neutrality, Accident, and Function*, pages 101–134. Oxford University Press.
- Jones, C. G., Lawton, J. H., and Shachak, M. (1997). Positive and negative effects of organisms as physical ecosystem engineers. *Ecology*, 78(7):1946–1957.
- Ray, T. S. (1991). An approach to the synthesis of life. In Langton, C., Taylor, C., Farmer, J., and Rasmussen, S., editors, *Artificial Life II*, volume X of *SFI Studies in the Sciences of Complexity*, pages 371–408. Addison-Wesley.
- Schuster, P. (2011). Mathematical modeling of evolution. solved and open problems. *Theory in Biosciences*, 130(1):71–89.
- Taylor, T. (2013). Evolution in virtual worlds. In Grimshaw, M., editor, *The Oxford Handbook of Virtuality*. Oxford University Press. (forthcoming).
- Van Valen, L. (1973). A new evolutionary law. *Evolutionary Theory*, 1:1–30.
- von Neumann, J. (1966). *The Theory of Self-Reproducing Automata*. University of Illinois Press, Urbana, Ill.
- Waddington, C. (1969). Paradigm for an evolutionary process. In Waddington, C., editor, *Towards a Theoretical Biology*, volume 2, pages 106–128. Edinburgh University Press.

Size Does Matter: The Impact of Size on Hoarding Behaviour

Olaf Witkowski and Nathanael Aubert

University of Tokyo, Japan
 olaf@is.s.u-tokyo.ac.jp
 nathanael.aubert@is.s.u-tokyo.ac.jp

Extended Abstract

Food-hoarding behaviour has been shown to be a viable, adaptive behaviour (Andersson and Krebs, 1978; Smulders, 1998). A significant amount of effort has been made to understand pilferage control and tolerance (Clarke and Kramer, 1994; Vander Wall and Jenkins, 2003; Ekman et al., 1996). In addition, many of those models study cache spacing (Kraus, 1983) and collective hoarding (AV and MY, 1991; Brodin and Ekman, 1994). However, Andersson and Krebs (1978) show that reciprocal pilfering can make hoarding systems resilient to invasions of cheaters, and argue that the hoarding behaviour does not need to be considered as an altruistic mechanism. Most of the research on food-hoarding has disregarded the influence of primary factors such as distribution of food over time or the consequences of agents size on their caching behaviour. This is the point where the modelling facet of Artificial Life may bring new highlights on hoarding behaviour.

This paper investigates the impact of changes in environment resources, available to a population of individuals, on their caching strategy. To do so, we present a simple agent-based model incorporating a population of individuals capable of storing resources, adapting their behaviour through generations, in a world offering a differentiated cyclic food distribution.

Our model is based on agents striving to obtain food from the environment. They are given five possible actions: either eat, forage, store food, reproduce or do nothing. The decision mechanism is implemented by an artificial neural network with inputs set to food availability (“temperature”), current energy of the agent (“hunger”) and result of the last forage. We also feedback the results of the cached layer to give the agent some kind of memory. The weights of the neural network, randomized at first, are refined through mutations and crossovers on the span of multiple generations. The genotype also determines the agent’s size, that influences the cost of its actions. Agents are only interacting indirectly, through food availability. Every action having its defined cost, the choice of the agents to hoard collected resources is made at the expense of an extra cost in energy. Other factors, such as pilfering, guarding or recaching, are abstracted to action costs.

In this paper, we aim to identify a number of behaviours resulting from the variation of environment conditions in a minimalist agent-based model. Our first research hypothesis was that in the emergence of hoarding behaviour when winters get more arduous, that is when agents need to survive longer periods of time on restricted supply of resources.

In a first attempt to exhibit this phenomenon, we first simulated *gentle winters*, during which the food was enough for individuals to survive on it. We observe that after 30 to 40 generations, hoarding behaviour is completely discarded in favour of scavenging for food as much as possible even during winter, and reproducing during summer. In the case of gentle winters, the population curve fits closely the food availability (see figure 1, left), whereas tougher winters force the agents to hoard in order to survive (see figure 1, right).

From there, we gradually made winters more deadly, with the food availability function effectively dropping to zero. In the simulation runs in which agents are able to survive a few more winters, we can rapidly observe a wide range of adapted sizes and behaviours. Progressively selected by increasingly difficult winters, we can observe the agents storing food and eating from their stores in periods where the food supply drops to lower values.

Furthermore, we find that hoarding behaviour depends on agent size. In general, the agents tend to evolve to a certain range of sizes (see figure 2, left) and perform hoarding to survive the increasingly difficult winters. However, if the agent’s size passes a certain threshold (approximately 20), it usually adopts a hibernation strategy during winters to save energy. Agents of size 10 to 20 tend to adopt a mix of both strategies.

Two controls were implemented: eternal winter (no food availability) and eternal summer (food availability always high). In the first case, agents are dying quickly as expected. In the second case, the hoarding behaviour is completely marginal, and sizes are almost evenly distributed, with a slight bias toward bigger agents. Since in real simulations smaller agent sizes were favored, this bias was dismissed as irrelevant.

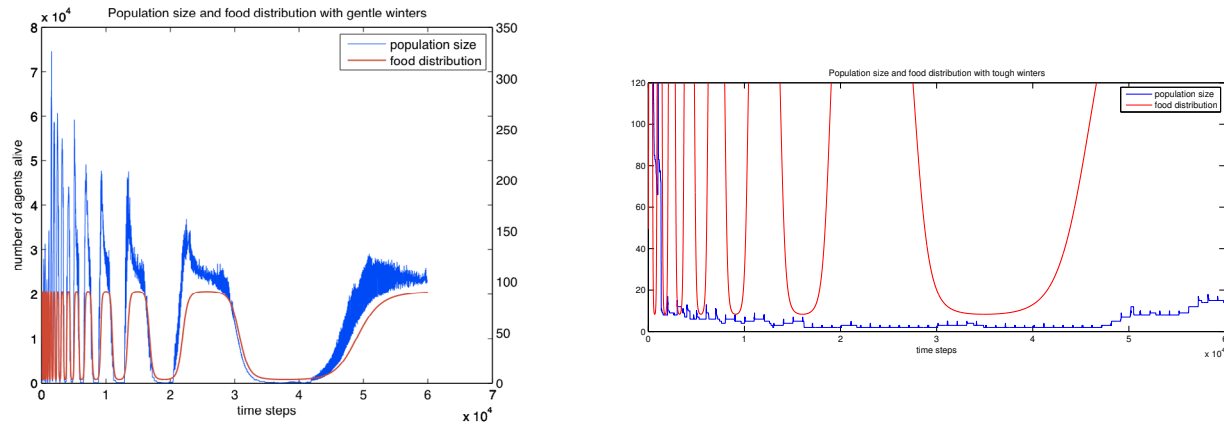


Figure 1: Comparative graphs of the population size and the food availability distribution through time with “gentle” winters (*left*), i.e. when the resources remains relatively abundant, and with “hard” winters (*right*), where the food is very rare.

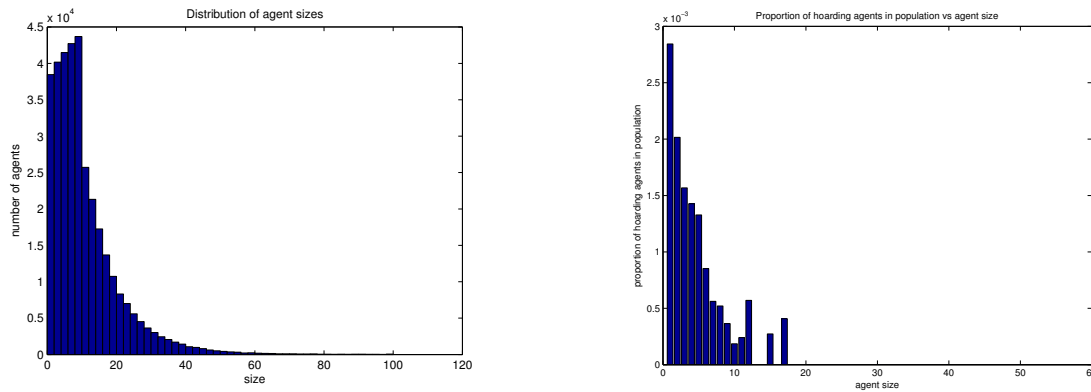


Figure 2: Number of individuals of each size (*left*). Proportion of agents of each size that exhibit hoarding behaviour (*right*)

Another behaviour also recurrently appears, when small agents take advantage of their cheap cost of reproduction, in order to produce as many offspring as possible. In mathematical biology, this strategy of survival focusing on the quantity of progeny over its quality, typically adopted by bacteria or insects, is referred to as an “r-strategy” (MacArthur and Wilson, 1967). The emergence of this so-called “r/K” opposition visibly demands no more than simplistic laboratory settings such as our model.

We are still investigating whether those hypotheses are compatible with other r/K characteristics, such as limited number of offspring. Besides, more action choices can be given to our agents, such as the ability to share food, in order to let more K behaviours emerge. Our results suggest that the agents’ size and the environment time cycles are major factors influencing their behaviour, as may be observed in real life. This also suggests that our model could somewhat predict behaviour modification to adapt to different conditions, such as abnormally long winters.

References

- Andersson, M. and Krebs, J. (1978). On the evolution of hoarding behaviour. *Animal Behaviour*, 26, Part 3(0):707 – 711.
- AV, B. and MY, M. (1991). Rate of plundering of reserves by tits: experimental investigations. *Soviet J Ecol*, pages 61: 322–336.
- Brodin, A. and Ekman, J. (1994). Benefits of food hoarding. *Nature*, 372(6506):510–510. 10.1038/372510a0.
- Clarke, M. F. and Kramer, D. L. (1994). The placement, recovery, and loss of scatter hoards by eastern chipmunks, *tamias striatus*. *Behavioral Ecology*, 5(4):353–361.
- Ekman, J., Brodin, A., Bylin, A., and Sklepovych, B. (1996). Selfish long-term benefits of hoarding in the siberian jay. *Behavioral Ecology*, 7(2):140–144.
- Kraus, B. (1983). A test of the optimal-density model for seed scatterhoarding. *ECOLOGY*, 64(3):608–610.
- Smulders, T. (1998). A game theoretical model of the evolution of food hoarding: Applications to the paridae. *AMERICAN NATURALIST*, 151(4):356–366.
- Vander Wall, S. B. and Jenkins, S. H. (2003). Reciprocal pilferage and the evolution of food-hoarding behavior. *Behavioral Ecology*, 14(5):656–667.

When is Happy Hour: An Agent's Concept of Time

Olaf Witkowski¹, Geoff Nitschke¹, Takashi Ikegami¹

¹ Ikegami Laboratory, Interdisciplinary Studies Department, University of Tokyo, Japan
 olaf@sacral.c.u-tokyo.ac.jp, geoff@sacral.c.u-tokyo.ac.jp, ikeg@sacral.c.u-tokyo.ac.jp

Extended Abstract¹

In *Artificial Life*, agent based modeling is a popular synthetic approach that often studies the evolutionary conditions responsible for adaptive group behavior. For example, emergent social phenomena such as communication and cooperation have been studied using agent models with a spatial distribution of agents and resources (Parisi, 1997), (Arita and Koyama, 1998). However, few studies have focused on the evolution of abstract concepts, such as a *concept of time*, that benefits individual and group behavior. In this study, agents attain a concept of time via learning to benefit from periodicity (cyclic resource growth) in the environment. Notable exceptions include the study of how memory extends an agents temporal horizon and increase its adaptability (Ching Ho et al., 2008). Nehaniv (1999) discusses the concept of narrative intelligence in temporally grounded agents. For example, the impact that stories of the past have upon an agent group's social behavior. In related work, Nehaniv et al. (2002) describe an information-theoretic model for individual and social learning in temporally grounded agents. The capacity to learn from environmental temporal patterns such as periodicity is beneficial to a broad spectrum of organisms, from Amoebae (Saigusa et al., 2008) to human civilizations (Hassan, 1997). This study investigates how an evolved sense of time can be used to adapt agent group behavior. The objective is to use a minimalist simulation model (with a spatial distribution of food and agents) to demonstrate that learning a concept of time facilitates efficient group foraging behavior. The concept of time is embedded into agent signals (indirectly indicating distances to food), and environmental behavior (*seasonal variations* define when food is scarce versus plentiful). Each agent is defined by a *local clock* (it's lifetime), and the environment by a *global clock* (oscillations of resource growth). The hypothesis is that resource growth cycles coupled with agent signaling about resource locations are sufficient conditions for agents to increase the efficiency of group foraging behavior. That is, agents adapt their behavior to exploit altruistic signals, learning when food is plentiful versus when it is not.

Figure 1 presents an example of the environment (left) and the agent *Artificial Neural Network* (ANN) controller (right). Controllers are adapted with an *Evolutionary Algorithm* (EA) that evolves connection weights. Agent fitness equals the food amount consumed during a lifetime. Agents consume U energy units for standing still, and $U + W$ energy units for moving. The EA selects for agent behaviors that stop and conserve energy when food is scarce, and behaviors that cause agents to move about foraging when food is plentiful. The environment is a two dimensional torus consisting of P evenly spaced food patches, governed by cyclic periods of food abundance (*summer*) and scarcity (*winter*). Each iteration, agents (speakers) emit a signal that conveys how many iterations in the past the speaker was on a food patch. From this, receivers (closest agents) learn that a food patch is Y grid spaces away in a given direction (agents receive signals from both directions).

To test the hypothesis that agent groups learn to use the concept of time, a comparative study was conducted. Experiments were executed where agent signalling was *switched on* and *switched off*. Results indicated that agents evolved a meaningful association between signals, cyclic resource growth, and foraging behavior. That is, agents interpret signals differently given different *seasons*, and adapt foraging behavior based on signals received. When there are few resources, agents signal that food has not been eaten (on average) in a long time. This causes agents to conserve energy by moving less. Where as, when resources are plentiful, agents signal that food has been eaten (on average) recently, causing agents to be less energy conservative.

Figure 2 depicts agent controller average hidden layer activation versus signal intensity in *winter* and *summer*. The broad spread of average internal state values is indicative of frequent agent activity in the summer (figure 2, *second figure*). Where as, the relatively compact clustering of average internal state values is indicative of infrequent agent activity in the winter (figure 2, *first figure*). Therefore, this signalling behavior indicates that agents effectively adapt their behavior to the environment's seasonal variation. In simulations without signalling, this disparity in average signal intensity and internal state values is not observed (figure 2, *third and fourth figures*). That is, approximate uniformity in the spread of activation values in winter and summer indicate that agents do not adapt their behavior to seasonal variation. Thus, in simulations including the concept of

¹This research was funded by a *Japan Society for Promotion of Science* fellowship for Foreign Post-Doctoral Researchers and a Grant-in-Aid for Specially Promoted Research.

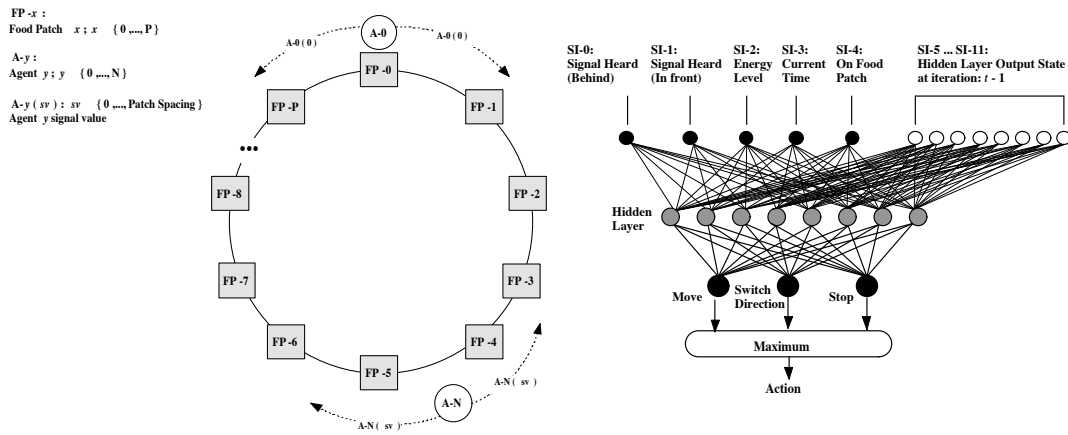


Figure 1: *Ring World Environment (Left)*: There are P , evenly spaced food patches, and N agents. Each iteration, agents emit signals indicating the time (number of iterations) since they were last on a food patch. Signals are broadcast in both directions, at a fixed number of grid spaces. *Right*: Agent controller is a recurrent feed-forward neural network. *SI*: Sensory Input.

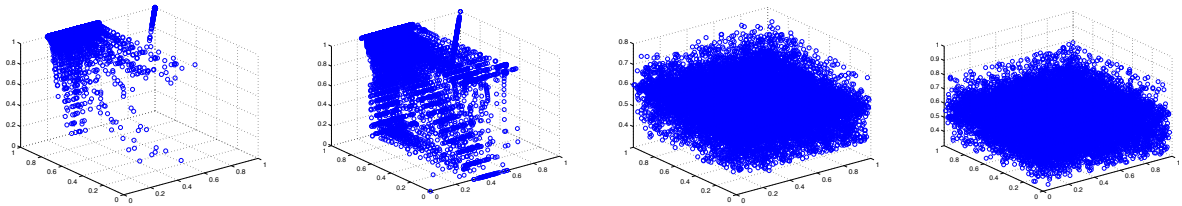


Figure 2: *Internal State of Fittest Agent with (first two plots), without signaling (last two plots)*: Average activation values plotted against signal values at generation 500. Average internal controller state is mapped for periods of food scarcity (left, both plots) and abundance (right, both plots). These periods are known as winter and summer, respectively.

time (signalling and cyclic resource growth), agents use signals sent under different environmental conditions in order to adapt foraging behavior and attain a higher fitness (compared to simulations where agents do not employ the concept of time).

It is important to note that the concept of time developed in this simulation is different from our notion of time. Thus, future work will investigate defining internal state mechanisms that indicate if agents have acquired a concept of time analogous to our own. Furthermore, we will examine interactions between agents' *local clocks* (for example, each agent's notion of when different seasons occur), and the environment's *global clock* (defining the periodicity of seasons), and if the synchronization of local and global clocks facilitates beneficial adaptive behavior in agent groups.

References

- Arita, T. and Koyama, Y. (1998). Evolution of linguistic diversity in a simple communication system. *Artificial Life*, 4(4):109–124.
- Ching Ho, W., Dautenhahn, K., and Nehaniv, C. (2008). Computational memory architectures for autobiographic agents interacting in a complex virtual environment: a working model. *Connection Science*, 20(1):21–65.
- Hassan, F. (1997). The dynamics of a riverine civilization: A geoarchaeological perspective on the Nile valley, Egypt. *World Archaeology*, 29(1):51–74.
- Nehaniv, C. (1999). Narrative for artifacts: transcending context and self. In *Proceedings of the 1999 AAAI fall symposium*, pages 101–104, Los Alamitos, USA. AAAI Press.
- Nehaniv, C., Polani, D., and Dautenhahn, K. (2002). Meaningful information, sensor evolution, and the temporal horizon of embodied organisms. In *Proceedings of Artificial Life VIII*, pages 345–349, Los Alamitos, USA. MIT Press.
- Parisi, D. (1997). An artificial life approach to language. *Mind and Language*, 59(1):121–146.
- Saigusa, T., Tero, A., and Nakagaki, T. (2008). Amoebae anticipate periodic events. *Physical Review Letters*, 100(1):18101–18104.

Fitness Proportionate Sharing: a Different Perspective for Co-evolution of Diverse Population.

Abraham Workineh, Abdollah Homaifar

Autonomous Control and Information Technology Center
North Carolina A & T State University, Greensboro, NC 27411

atworkin@ncat.edu, homaifar@ncat.edu

Extended Abstract

By the same analogy with natural ecosystems, niching methods tend to achieve a natural emergence of niches and species in the search space. Niching enables the standard Genetic Algorithm (GA) to discover multiple optima by forming subpopulations representing locally optimized solutions (Sareni and Krahenbuhl, 1998, Dick and Whigha, 2006). It provides a restoring force for the GA to counterbalance the impact of genetic drift due to the selection pressure. The traditional fitness sharing technique has a limitation when the multi-modal function has several unequal peaks, particularly when there is a large gap between the fitness values at the peaks (Mahfoud, 1994). It evolves the whole population towards convergence at the location of the highest peak unless a relatively large population size is used. We developed a novel niching technique based on fitness proportionate resource sharing to overcome this drawback. An analysis is made both using equations and simulations on well known multi-modal test functions with unequal peaks. Unlike the conventional fitness sharing scheme, the gap in fitness values of the peaks does not affect the performance of the proposed niching scheme. In our previous work (Workineh and Homaifar, 2012), we applied this niching technique for evolving hierarchical cooperation in learning classifier systems. This technique is based on the notion of limited resources where individuals in a given niche share the resource of that niche in proportion to their strength. The sharing function is given in equation (1) and the derated fitness of an individual i is given by equation (2).

$$sh(d_{i,j}) = \begin{cases} F_j & , \text{if } d_{i,j} < \sigma_{sh} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$F_{sh,i} = \frac{F_i}{\sum_{j=1}^M sh(d_{i,j})} \quad (2)$$

Where M is the subpopulation size at a given niche, d_{ij} is the phenotypic distance between individuals i and j .

Consider the function with 5 unequal peaks as shown in Figure 1 with fitness values at the peaks of P_1 to P_5 . And the subpopulation size at each of the niches is denoted by n_1 to n_5 respectively.

Using the traditional fitness sharing (Deb and Goldberg, 1989, Cioppa, et al, 2007), the shared fitness of an individual at the k^{th} niche is given by equation (3).

$$P'_k = \frac{P_k}{n_k} \quad (3)$$

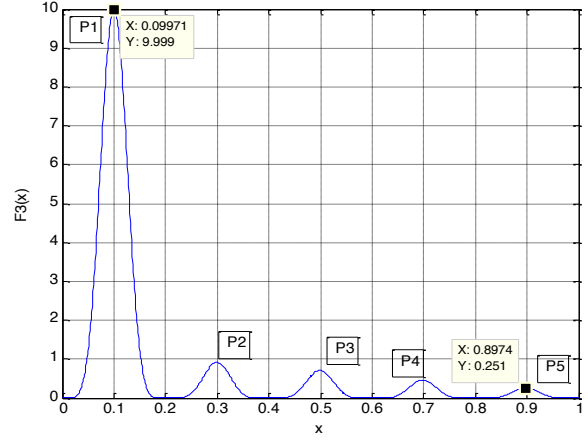


Figure 1:-A multi-modal function with 5 unequal peaks with a large fitness variation between the highest and lowest peaks.

Assuming that after sufficient iteration almost all the population distributes around the five peaks, we get equation (4).

$$n_1 + n_2 + n_3 + n_4 + n_5 = N \quad (4)$$

To discover all the peaks, it is required that the shared fitness values at each niche should be approximately equal (i.e. $P'_1 = P'_2 = P'_3 = P'_4 = P'_5$).

Substituting and rearranging terms, the number of individuals at the k^{th} niche is governed by equation (5).

$$n_k = \frac{P_k}{\sum_{i=1}^5 P_i} * N \quad (5)$$

If a niche size of at least two individuals is required at the lowest peak (i.e. $n_5 \geq 2$), the minimum population size required to discover all the peaks using the traditional sharing technique is given by equation (6).

$$N \geq 2 * \left(\frac{\sum_{k=1}^5 P_k}{P_5} \right) \quad (6)$$

This indicates that when the objective function has unequal peaks, the traditional sharing scheme has a threshold requirement on the minimum population size to discover all the peaks. As the gap of the peak values increases, the required minimum population size also increases drastically.

But using our method, the shared fitness of an individual at the

k^{th} niche is given by equation (7).

$$P'_k = \frac{P_k}{\sum_{i=1}^n P_i} \quad (7)$$

Where n_k is the subpopulation size at the k^{th} niche (location of a peak). After sufficient iteration, individuals in the same niche will have approximately equal fitness (i.e. $f_i=f_j$, for two individuals i and j). Hence equation (7) can be simplified as in equation (8).

$$P'_k = \frac{1}{n_k} \quad (8)$$

From equation (8), for the shared fitness values to be equal, the population has to be evenly distributed among all the peaks, irrespective of the difference in the fitness value at the peaks (i.e. $n_1=n_2=n_3= n_4= n_5=N/5$). In general, for a multimodal function having M optimum points, the expected number of individuals at the k^{th} peak using the traditional fitness sharing and FPN is given by equation (9) & (10) respectively.

$$n_k = \frac{F_k}{\sum_{i=1}^M F_i} * N \quad (9)$$

$$n_i = \frac{N}{M} \quad (10)$$

Where F_k is the fitness value representing the k^{th} niche, and N is the population size.

In the results and table shown below, TN and FPN refer to the traditional and our sharing methods respectively. Figure 2 depicts a comparison using simulation between FPN and TN for the given multimodal function vs population size. Table 1 shows the distribution of the population (averaged over 10 runs) among the various peaks for the same function. As can be seen from the table, FPN distributes the population among the various peaks uniformly irrespective of the fitness difference at the peaks.

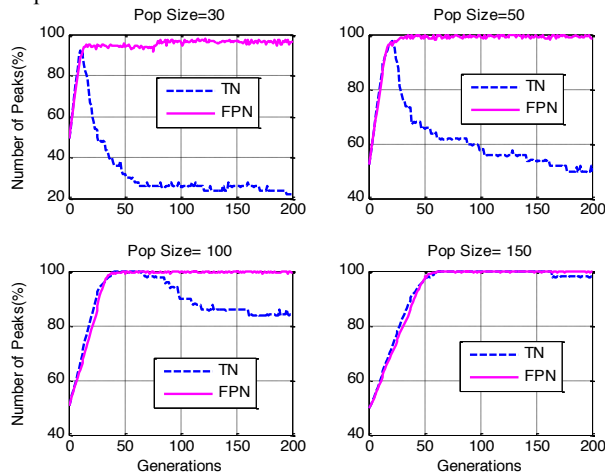


Figure 2: Percentage of number of peaks discovered for the function given in Figure 1, averaged over 10 runs.

Table 1: Population distribution at the five different peaks, averaged over 10 runs for $F_3(x)$.

PS		Peak1	Peak2	Peak3	Peak4	Peak5
30	TN	29.8	0.2	0	0	0
	FPN	7.9	6.9	6.3	5.5	3
50	TN	47.1	2	0.7	0.1	0
	FPN	12.1	10.7	9.7	9.3	7.5
100	TN	88	5.7	4.5	1.4	0.2
	FPN	21.6	20.6	20	19.9	17.1
150	TN	128.4	9.8	7	3.5	1.3
	FPN	31.9	31.5	30.5	28.9	26.9

Key Words: Evolutionary algorithm, fitness sharing, genetic algorithms, multimodal optimization, niching methods.

ACKNOWLEDGEMENTS

This material is based in part upon work supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454.

REFERENCES

- Cioppa, A.D., etal (2007), "Where are the Niches? Dynamic Fitness Sharing", IEEE Transactions on Evolutionary Computation, Vol. 11, No. 4.
- Deb, K. and Goldberg, D.E.(1989) "An investigation of niche and species formation in genetic function optimization," in *Proc. 3rd Int. Conf.Genetic Algorithms*, J. D. Schaffer, Ed. San Mateo, CA: Morgan Kaufmann, pp. 42-50.
- Dick, G. and Whigha, P.A.(2006), "Spatially-Structured Evolutionary Algorithms and Sharing: Do They Mix?", Proceedings of Proceedings of the 5th International Conference on Simulated Evolution and Learning (SEAL), pp. 457-464.
- Mahfoud, S.W.(1994), "Crossover Interaction Among Niches," Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, pps 188-193.
- Sareni, B. and Krahenbuhl, L.(1998)," Fitness Sharing and Niching Methods Revisited", IEEE Transactions on Evolutionary Computations, Vol. 2, Issue 3, pps.97-106.
- Workineh, A. and Homaifar, A. (2012, submitted), "Evolving Hierarchical Cooperation in Classifiers via Fitness Proportionate Niching", IEEE Transactions on Evolutionary Computations.

Collective Dynamics

Extended Abstracts

Explicit and Implicit Directional Information Transfer in Collective Motion

E. Ferrante^{1,2}, A. E. Turgut^{1,2}, C. Huepe³, M. Birattari¹, M. Dorigo¹ and T. Wenseleers²

¹IRIDIA, CoDE, Université Libre de Bruxelles, 50 Av. Franklin Roosevelt CP 194/6, 1050 Brussels, Belgium

²Laboratory for Entomology, Katholieke Universiteit Leuven, 59 Naamsestraat - bus 2466, 3000 Leuven, Belgium

³Northwestern Institute on Complex Systems, Northwestern University, Evanston, IL 60208, USA

Extended Abstract

We study the cohesive coordinated collective motion of a group of mobile autonomous robots. We use virtual interactions between robots implemented via proximal control, which allows the robots to reach a stable formation using virtual potential functions (Turgut et al., 2008; Ferrante et al., 2011). The alignment component can be seen as a mechanism for directional information transfer (Sumpter et al., 2008). We refer here to information transfer in collective motion as the process through which robot orientation is transferred to its neighbors over time.

We consider here two information transfer mechanisms for collective motion in a group of mobile robots. The first one exploits information transfer through direct communication and requires robots equipped with proximity, orientation sensing and communication devices. We propose communication strategies that allow the robots informed about a desired direction of motion to influence the rest of the group (Couzin et al., 2005; Ferrante et al., 2011). The second mechanism consists of information transfer without the alignment component and communication (Ferrante et al., 2012), which can be used on simpler robots only equipped with proximity sensors. We developed a simple motion control mechanism that allows a group of robots to perform collective motion in a random direction without needing robots informed about a desired direction or an explicit alignment behavior: information among the robots is thus transferred indirectly.

Information transfer via communication We consider a case where some robots have a persistent desired direction of motion (desired direction A) which could, for example, represent the direction to a food source. There is also a second desired direction (desired direction B), only present during a time window which could, for example, represent the escape direction from a predator. Desired direction B is in conflict with A : it points in the opposite direction and has higher priority. The objective is to move the group in the direction that, at a given time, has the maximum priority, and to keep the group cohesive.

We proposed a self-adaptive communication strategy (SCS), that is an extension of two previously proposed strategies (Ferrante et al., 2011). In SCS, the robot sends an angle θ_{s_0} and receives angles θ_{s_i} from its k neighbors. It computes the average of the received angles: $\mathbf{h} = \frac{\sum_{i=0}^k e^{j\theta_{s_i}}}{\|\sum_{i=0}^k e^{j\theta_{s_i}}\|}$. The angle sent is: $\theta_{s_0} = \angle [w\mathbf{g} + (1-w)\mathbf{h}]$. The parameter $w \in [0, 1]$ is the degree of confidence of the robot on the desired direction \mathbf{g} . Non-informed robots use $w = 0$ (they possess no information about \mathbf{g}). Robots informed about desired direction B use $w = 1$, which makes them stubborn. Robots informed about desired direction A increase w when they measure high level of consensus in the information received by the neighbors, and decrease it otherwise.

Figure 1a shows the distribution of the accuracy over time, which measures how close the group direction is to desired direction A . In these experiments, 1% of the robots is always informed about desired direction A . During the time window where an additional 1% of the robots is informed about desired direction B , the accuracy reaching 0 indicates that desired direction B is being followed. In the remaining part of the experiment, the group correctly follows desired direction A . This result has been validated on real robot experiments (Fig. 1b). In addition, we show that SCS results either in a better accuracy (Fig. 1a and Fig. 1b) or in a better group cohesion (Fig. 1c) than two previously proposed strategies, HCS and ICS. The full results are reported in Ferrante et al. (2011).

Information transfer without communication We study information transfer with no alignment behavior and no communication. Our approach is based on a novel Magnitude Dependent Motion Control (MDMC) method, used to compute the forward and angular speed of the robot. The two speeds depend on the magnitude and angle of \mathbf{f} , the vector resulting from proximal control that encodes the attraction and repulsion strength from the neighbors. \mathbf{f}_x and \mathbf{f}_y denote the projection of \mathbf{f} on the axis parallel (x) and perpendicular (y) to the direction of motion of the robot. In MDMC, the forward speed u is proportional to the x component: $u = K_1 \mathbf{f}_x + U$, and the angular speed ω to the y component: $\omega = K_2 \mathbf{f}_y$, where U is a forward biasing speed.

Figure 1 (second row) shows the results of experiments performed with simulated and real robots. MDMC is compared to the method used in Turgut et al. (2008): Magnitude Independent Motion Control (MIMC). In MIMC, the forward and angular speed do not depend on the magnitude of the vector \mathbf{f} but just on its angle. Figure 1d shows the distribution of the order metric over time, which measures the degree of alignment in the group. MDMC achieves ordered motion without the alignment behavior

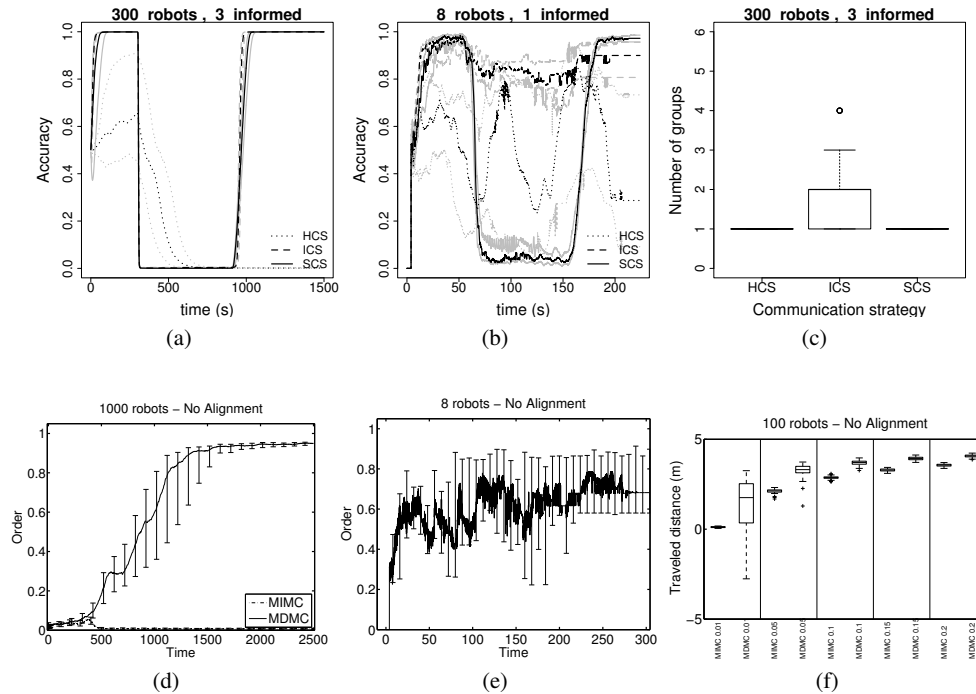


Figure 1: Experiments with simulated and real robots. Time dependent data is sampled every second. Black lines are the medians of the distribution, whereas grey lines (in (a), (b)) and error bars (in (d), (e)) represent the 25% and the 75% quartiles.

and without informed robots, whereas MIMC does requires informed robots or the alignment behavior. These conclusions are backed up by real robot experiments (Fig. 1e). Moreover, when a proportion of informed robots (0.01, 0.05, 0.1, 0.15, 0.2 as indicated in the plot) is introduced, the group is able to travel further along a desired direction of motion using MDMC than using the earlier MIMC method (Fig. 1f).

Discussion and conclusion We showed that the information needed to achieve collective motion can be transferred either directly or indirectly. Direct information transfer requires robots with orientation sensing and communication devices. We developed a communication strategy that can cope with two conflicting desired directions of motion. We also proposed a novel mechanism for robot motion that exploits indirect information transfer. This allows robots that lack the above mentioned capabilities to perform cohesive collective motion without communication, showing that implicit information transfer on the heading direction takes place even without communication. In future work, we will use information-theoretic metrics to measure information transfer more rigorously.

Acknowledgements This work was partially supported by: the European Union (ERC Advanced Grant “E-SWARM”, contract 246939); the F.R.S.-FNRS of Belgium’s French Community (Meta-X project); the Vlaanderen Research Foundation Flanders (H2Swarm project), the US National Science Foundation (Grant No. PHY-0848755).

References

- Couzin, I. D., Krause, J., Franks, N. R., and Levin, S. A. (2005). Effective leadership and decision-making in animal groups on the move. *Nature*, 433:513–516.
- Ferrante, E., Turgut, A. E., Huepe, C., Stranieri, A., Pinciroli, C., and Dorigo, M. (2012). Self-organized flocking with a mobile robot swarm: a novel motion control method. IridiaTr2012-003, Université Libre de Bruxelles, Belgium.
- Ferrante, E., Turgut, A. E., Stranieri, A., Pinciroli, C., Birattari, M., and Dorigo, M. (2011). A self-adaptive communication strategy for flocking in stationary and non-stationary environments. IridiaTr2012-002, Université Libre de Bruxelles, Belgium.
- Sumpter, D. J. T., Buhl, J., Biro, D., and Couzin, I. D. (2008). Information transfer in moving animal groups. *Theory in Biosciences*, 127(2):177–186.
- Turgut, A. E., Çelikkanat, H., Gökçe, F., and Şahin, E. (2008). Self-organized flocking in mobile robot swarms. *Swarm Intelligence*, 2(2):97–120.

Diverse Behaviors in Swarm Robotics with Novelty Search

Jorge Gomes¹, Paulo Urbano¹ and Anders Lyhne Christensen²

¹LabMAG – Faculdade de Ciências da Universidade de Lisboa, Portugal

²Instituto de Telecomunicações & Instituto Universitário de Lisboa (ISCTE-IUL), Lisboa, Portugal
jgomes@di.fc.ul.pt, pub@di.fc.ul.pt, anders.christensen@iscte.pt

Extended Abstract

Novelty search (Lehman, 2011a) is a divergent evolutionary technique that drives evolution towards behavioral diversity instead of maximizing a fitness function. Novelty search has been shown capable of finding numerous different classes of solutions in a single evolutionary run (Lehman, 2001b), as opposed to fitness-based evolution, where a particular run typically converges in a single solution. In our work, we apply novelty search to the evolution of neurocontrollers for a swarm of simulated robots that must perform an aggregation task. We show that novelty search is able to find a broad diversity of swarm behaviors that can solve the task.

The main idea behind novelty search is to reward novel solutions instead of progress towards a well-defined goal. In novelty search, the fitness function is replaced with a novelty metric that measures how different a solution is from other previously evolved solutions. The novelty of a newly generated solution is computed with respect to the behaviors in an archive of past solutions and to the current population. The archive is initially empty. During evolution, a solution is added to the archive if it is significantly different from the ones already there. In this way, the archive is a representation of the explored behavior space. To measure the novelty of a given point in the behavior space, the average distance the k -nearest neighbors of that point is calculated, where k is a parameter of the algorithm. Candidates from more sparse regions of the behavior space thus receive higher novelty scores, thereby creating a constant pressure to evolve solutions with novel behavioral features.

We use an aggregation task for our experiments. In this task, the robots should move around in a bounded arena to search for each other and ultimately form a single aggregate. We made the task more challenging by increasing the size of the arena and by reducing the sensors capabilities, compared to previous studies on aggregation in robots (see for instance (Trianni, 2003)). We experimented with fitness-based evolution and with novelty search with two distinct novelty measures. The fitness function used to drive the fitness-guided evolution is based on the average distance of the robots to their collective center of mass. The average distance is sampled throughout the simulation at regular intervals, and these samples are then combined in a single fitness value using a weighted average that gives more weight to the measurements taken towards the end of the simulation. The two novelty measures used in the novelty search experiments are: (1) The average distance of the robots to the center of mass of the swarm sampled over time; and (2) the number of robot clusters sampled over time.

Our previous results (Gomes, 2012) showed that there was no significant difference between the fitness trajectories obtained in fitness-based evolution and in novelty search. However, we found significant differences in the exploration of the behavior space. To analyze the exploration of the behavior space, we used Kohonen self-organizing maps (Kohonen, 1990). The Kohonen maps allowed us to map the high dimensionality of the behavior space (each behavior vector had a length of 50) to two dimensions, maintaining the distance relations between the behaviors. In both novelty search experiments (center of mass measure and number of clusters measure), novelty search successfully explored behavior zones that the fitness-based evolution could not reach. In the experiment with the center of mass measure in particular (Figure 1), we can see that the fitness-based evolution avoided the zones where the average distance to the center of mass rises beyond the initial value. Fitness-based evolution thus bypassed a class of good solutions, namely those in which the robots navigate along the walls to find one another. Novelty search, on the other hand, explored the behavior space more uniformly.

Analyzing the evolved controllers in action, the differences are also noticeable. In the fitness-based evolution, the highest scoring controllers were always very similar: the robots explore the environment in large circles, and form static clusters when they encounter one another. If the cluster is small, the robots abandon it after a while and start exploring again. Novelty search, on the other hand, found in each evolutionary run several controllers that could solve the task in different ways, that is, controllers that caused the robots to display distinct macroscopic swarm behaviors. With the aid of the Kohonen maps we were also able to identify the good and phenotypically distinct controllers, by observing the behavior of the highest scoring solutions (in terms of fitness) mapped to each neuron.

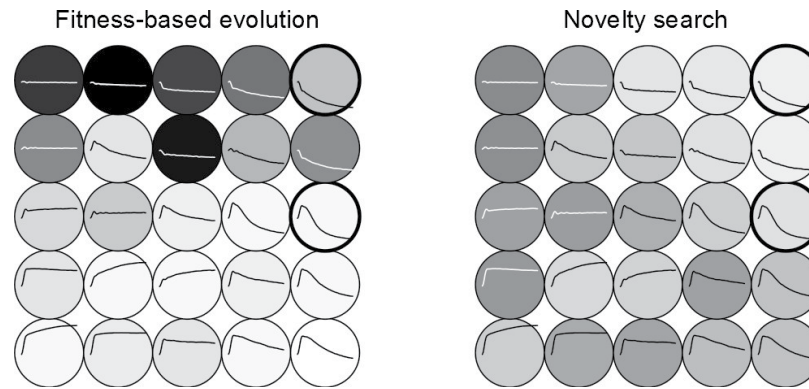


Figure 1. Kohonen maps representing the explored behavior space in fitness-based evolution (left) and in novelty search (right). Each circle represents a behavior zone characterized by the average distance of the robots to the center of mass of the swarm over time (depicted by the plot inside each circle). The darker the background of a circle is, the more that zone was explored. The circles corresponding to the best behaviors (the ones with a low average distance to the center of mass at the end) have a bold circle.

Using the center of mass novelty measure, the following good behaviors were evolved: (1) The robots go straight forward until they hit a wall, and then, depending on the angle of approach, they stay there for a while or start moving along the wall until they find other robots; (2) Similar to (1), but when they meet each other they continue to follow the wall until they hit a corner, aggregating there; (3) Similar to the behavior evolved by fitness, but without splitting the small clusters; (4) Similar to (3) but navigating in the environment only in straight trajectories instead of curves. With the number of clusters novelty measure, different solutions were found: (1) The robots go towards walls, navigate along them and when they find another robot, they form a single file, keeping a fixed distance; (2) The robots navigate in circles in the environment, forming a static cluster when they meet each other; (3) Similar to (2), but they randomly abandon their respective clusters; (4) They navigate in circles and when two robots meet at some distance, one tries to follow the other. When robots collide, they form a cluster and remain aggregated.

We have shown that novelty search can be used to find diversity of solutions in the swarm robotics domain. The outcome of our study is consistent with the results described in previous works on applying novelty search to other domains. Our results suggest that novelty search could find a greater behavioral diversity than fitness-based evolution because (i) each fitness-based evolution focused in a single class of solutions, and (ii) the stepping stones necessary to reach certain types of behaviors were penalized by the fitness function. We demonstrated that the fitness-based evolution, while searching exclusively for the best solution, can bypass other interesting and equally good solutions to the task. However, the behavior space that is explored in novelty search is closely related to the novelty measure that is used. In our experiments, each measure could find a broad diversity of behaviors, but the behaviors found by each measure were different. This is not an unexpected result, but has important implications. On one hand, it means that even more diversity of solutions can be found by experimenting with different novelty measures. On the other hand, if a poor measure is defined, the evolution may fail to find solutions to the problem.

We argue that the drive of novelty search towards behavioral diversity is especially useful in swarm robotics. Novelty search can generate a diversity of effective robot controllers in a single evolutionary run, as opposed to the fitness-based evolution, in which a particular run often converges to a single solution. This diversity can provide a range of different solutions to the experimenter who is running the evolutionary process. This is especially relevant in the domain of swarm robotics, because the dynamical interactions between the robots and the environment may result in many behavioral possibilities (Trianni, 2006). Novelty search can explore these possibilities, potentially revealing new and unexpected forms of self-organization.

References

- Gomes, J., Urbano, P., Christensen, A. L. (2012). Introducing Novelty Search in Evolutionary Swarm Robotics. To appear in *8th International Conference on Swarm Intelligence*, Springer Verlag, Berlin, Germany, in press.
- Lehman, J., Stanley, K.O. (2011a). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189-223.
- Lehman, J., Stanley, K.O. (2011b). Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of Genetic and Evolutionary Computation Conference*, pages 211-218. ACM, New York, NY.
- Trianni, V., Groß, R., Labella, T.H., Sahin, E., Dorigo, M. (2003). Evolving aggregation behaviors in a swarm of robots. In *Advances in Artificial Life: 7th European Conference - ECAL 2003*, pages 865-874. Springer Verlag, Berlin, Germany.
- Trianni, V. (2006). *On the Evolution of Self-Organising Behaviours in a Swarm of Autonomous Robots*. Ph.D. thesis, Université Libre de Bruxelles, Brussels, Belgium.
- Kohonen, T. (1990). The self-organizing map. *Proc. of the IEEE*, 78(9) :1464-1480.

Computational Neuroecology of Communicated Somatic Markers

Kyle I. Harrington¹, Megan M. Olsen² and Hava T. Siegelmann³

¹Michtom School of Computer Science, Brandeis University, Waltham, MA 02454

²Department of Computer Science, Loyola University Maryland, Baltimore, MD 21210

³Department of Computer Science, University of Massachusetts, Amherst, MA 01003

Corresponding author: kyleh@cs.brandeis.edu

Extended Abstract

Introduction

Certain features of physiology (hunger, hormones, heart rate, etc.) and representations of physiology within the brain are somatic markers that influence behavior and decision making (Damasio et al., 1996; Bechara et al., 2000). Computationally modeling the neural bases of behavior is a goal of computational neuroethology (Beer and Chiel, 2008). Studies in computational neuroethology account for neural mechanisms, biomechanics, and ecological context, but generally focus on an individual.

Neuroecology studies social behaviors and their relationship to neural attributes. For example, the larger hippocampus of the male meadow vole who maintains a larger home range requires additional spatial ability (Sherry, 2006). The distinction between neuroethology and neuroecology arise from neuroecology's study of the linkage between stimulus, neural processes, behavior, and the corresponding effects on population and community (Zimmer and Derby, 2011).

The somatic marker hypothesis offers a physiological basis for emotion. Somatic markers are thought to stem from basic survival behaviors, and it has been hypothesized that emotional communication can increase the survival rate of a population. We investigate these neuroecological questions in predator-prey simulations by exploring the effect of communicated somatic markers on individuals and their ecology in order to establish an understanding of their evolvability. We previously explored the benefits of communicated somatic markers for the species and individual (Harrington et al., 2011), and now examine the effects of somatic markers on individuals and ecologies. Our findings support selective favorability of communicated somatic markers; in particular, we show how fear, happiness, and to a lesser extent surprise, can be favored by natural selection.

Model

Our multi-species agent-based model based upon (Harrington et al., 2011) is a torus inhabited by three species related by predator-prey interactions: rabbits, foxes, and carrots. Foxes feed on rabbits, while rabbits feed on carrots. Carrots serve as both an energy input and a vector of disease for the

Emotion	#	Experience
Happiness	1	1 if ate food, 0 otherwise
	2	1 if reproduced, 0 otherwise
Fear	1	number of neighboring predators
	2	1 if self will starve next turn, 0 otherwise
Anger	1	e^{hunger}
	2	$hunger / \text{starvation limit}$
Disgust	1	1 if ate diseased food, 0 otherwise
	2	fraction of diseased neighboring conspecifics
Sadness	1	time since last reproduction
	2	the decrease in number of surrounding foods, if applicable; 0 otherwise
Surprise ¹	1	$\frac{\sum_e E_e(t, x, y) - E_e(t-1, x, y)}{5}$
	2	$\frac{\sum_e (\tanh(E_e(t, x, y) / E_e(t-1, x, y)) + 1) / 2}{5}$

Table 1: Somatic markers used for emotional response. Rabbits use either #1 or #2, whereas foxes only use #2.

system. All entities breed while non-carrots also move, eat, experience hunger, and suffer from disease. For a detailed description of the model see (Harrington et al., 2011).

Results

We compare the effects of individual somatic markers in rabbits, comparing two definitions² of each somatic marker (Tbl. 1) when foxes do and do not use emotions. Figures show trials separated by configuration of rabbit emotion. Error bars represent the standard error centered around the mean as recorded during $t \in [1000, 2500]$ for 25 runs.

Fig. 1(a) and 1(b) show the fox and rabbit average ages. The fox average age increases dramatically when foxes use emotion. However, when only rabbits use emotion the average fox age is equivalent to neither species using emotion. When both rabbits and foxes use emotion the average fox age generally decreases when compared to only foxes using emotion, particularly for fear and anger. Average rabbit age only decreases with both species using anger, happiness, or surprise(a), as well as with rabbits using happiness₁.

¹Sums are taken over all emotions except surprise.

²When evaluating surprise definitions all emotions are activated and two tests are performed: (a) all emotions other than surprise are definition 1, and (b) all emotions other than surprise are definition 2. In both cases, surprise is tested with each of its own markers.

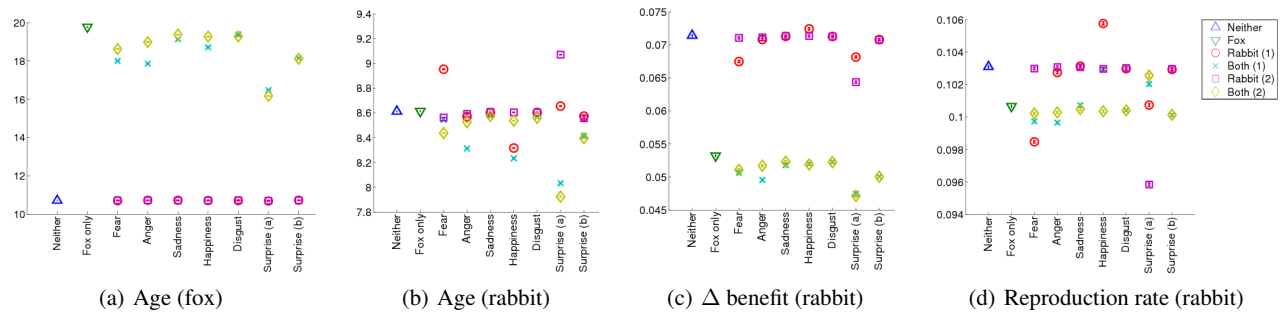


Figure 1: Rabbit and fox averages. X-axis shows the emotion being studied; in the case of surprise, all emotions are activated. The legend shows which somatic markers are in use for each series; for surprise, the number only corresponds to the somatic marker for surprise.

Both fox and rabbit populations (not shown) fall into two categories: high and low. There are more foxes when neither species uses emotion, or when only rabbits use emotion. Rabbit population sizes follow the opposite trend. The decrease in fox population when both use emotion is most likely because their improved knowledge allows them to be more effective hunters. This seems counter-intuitive given that it also correlates with a larger population of rabbits; however, the results of average age further support this idea.

The change in benefit of surroundings for a rabbit is

$$\Delta \text{benefit} = \Delta \text{neighboring carrots} - \Delta \text{neighboring foxes}.$$

Fox emotion correlates with a decrease in benefit (Fig. 1(c)). When only rabbits use emotion the benefit is generally near the baseline. However, fear₁ correlates with a decrease in the average benefit of rabbits. Given that this definition of fear correlates with an increase in the average rabbit age one would suspect that fear₁ causes rabbits to leave areas that are more abundant in food in favor of escaping predation.

Fig. 1(d) shows the reproduction rate ($R(t)$) change as a function of emotional configuration. When only rabbits use one emotion $R(t)$ is around baseline except in the case of *fear*₁ (significant decrease) and *happiness*₁ (significant increase). The decrease due to fear is due to high levels of fear halting reproduction. The increase due to happiness₁ correlates with decreases in average rabbit age described above.

Surprise(a) trials show a decrease in average benefit of surroundings (Fig. 1(c)), and an inversion of the effect of emotionally intelligent foxes on average rabbit reproduction rates (Fig. 1(d)). As in the other discussed cases, the use of either somatic marker definition for surprise only affects surprise(a) (when all other emotions only use somatic marker definition 1) and not surprise(b). This leads to the consideration that the synergistic effect of definition 1 somatic markers is not as simple as a linear combination of all active somatic markers. We recommend a more extensive study of the effect of secondary emotions such as surprise, employing many combinations of somatic markers to further our understanding of the nature of this non-linear combination.

Conclusion

We have shown that communicated somatic markers can correspond to individual benefits, whether those benefits are direct or secondary targets of natural selection. These findings suggest the selective favorability of communicated somatic markers. The communicated somatic marker utility in an ecology is a complex question. However, the relationship between the use of certain communicated somatic markers and objectives of natural selection, such as longevity and reproduction, suggests that understanding the origin of somatic markers is achievable by means of computational neuroecology as examined in this paper.

Acknowledgments

We thank Pinar Ozisik for many fruitful discussions. This work was supported in part by NSF #0757452, and ONR #109-0138R. Opinions, findings, and conclusions in this publication are those of the authors and do not necessarily reflect the views of the NSF or ONR. Computational support was provided by Brandeis HPC.

References

- Bechara, A., Damasio, H., and Damasio, A. (2000). Emotion, decision making and the orbitofrontal cortex. *Cerebral cortex*, 10(3):295.
- Beer, R. D. and Chiel, H. J. (2008). Computational neuroethology. *Scholarpedia*, 3(3):5307.
- Damasio, A., Everitt, B., and Bishop, D. (1996). The somatic marker hypothesis and the possible functions of the prefrontal cortex. *Philos. trans.: Bio. sci.*, 351(1346):1413–1420.
- Harrington, K., Olsen, M., and Siegelmann, H. (2011). Communicated somatic markers benefit both the individual and the species. In *2011 Intl. Joint Conf. on Neural Networks (IJCNN)*, pages 3272–3278.
- Sherry, D. (2006). Neuroecology. *Ann. Rev. Psychol.*, 57:167–197.
- Zimmer, R. and Derby, C. (2011). Neuroecology and the Need for Broader Synthesis. *Integrative and Comparative Bio.*, 51(5):751–755.

Evolutionary chasing between cooperators and defectors on the spatial prisoner's dilemma

Genki Ichinose¹, Masaya Saito¹ and Shinsuke Suzuki²

¹ Anan National College of Technology, 265 Aoki Minobayashi, Anan, Tokushima 774-0017, Japan

² Graduate School of Letters, Hokkaido University, Kita 10, Nishi 7, Kita-ku, Sapporo, Hokkaido, Japan
igenki@gmail.com

Extended Abstract

The role of migration in the evolution of cooperation has been discussed in spatial prisoner's dilemma games (PD). It is known that small but non-zero migration rates facilitate the formation and maintenance of cooperation. Several studies have dealt with the effect of migration on the evolution of cooperation (Enquist and Leimar, 1993; Sicardi et al., 2009; Vainstein et al., 2006; Janssen and Goldstone, 2006; Killingback et al., 2006; Ichinose and Arita, 2007, 2008; Pepper and Smuts, 2002; Pepper, 2007). Recently, Suzuki and Kimura (2011) found that oscillatory cooperation and defection dynamics take place if the migration rate is allowed to evolve. However, little is known about underlying mechanisms of the oscillatory dynamics. Moreover, most of these studies assumed random migration. In contrast, mobile organisms often move one place to another in a particular direction depending on their physical traits. Therefore, this paper deals with directional migration. Such an effect still has not been fully investigated in the context of evolution of cooperation.

In this study, we propose a spatial PD model in which each individual has the weighted probability of the directional migration in addition to the PD strategy. The evolutionary simulations resulted in evolutionary chasing between cooperators and defectors in a specific direction. Moreover, each of the local dynamics in this directional migration model was significantly different from those in the random migration. Figure 1 shows snap shots of a typical evolution. In this iteration, we realized the evolution of migration that was almost vertical. In the 931st generation, a cooperative cluster was generated (labeled "1"), and then spread to a lower space because they were mutually beneficial to each other (954th). Cooperators continued to move in the same direction, while some defectors emerged in the center of the cluster (976th). In the 997th generation, defectors moved in the same direction as the cooperators in order to catch them, and then were almost exploited (997th). In the same generation, another cooperative cluster was generated at the top space (labeled "2"). They moved downward (1019th, 1040th) but were exploited by defectors in the 1062nd generation. As a result, the #2 population collapsed (1086th). The #3 population then repeated the same cycle. These directional migrations were characterized by the entropy of the weighted direction probability. In addition, the rates of the global population extinction were reduced as a consequence of directional migration.

We showed that the collective behavior can affect the evolution of cooperation through reducing the extinction. Some studies have assumed the ability to cognitively detect other individuals, and it is known that cooperation can evolve in such situations. In contrast, without such knowledge, each individual collectively moves in the same direction in directional migration. Such behavior is observed at all levels of organisms including cells, individuals, ecosystems, and society. In some cases, the individual elements have no awareness of other elements. In such case, the collective movement is a universal mechanism in organisms with directional migration and contributes the population stability.

References

- Enquist, M. and Leimar, O. (1993). The evolution of cooperation in mobile organisms. *Animal Behavior*, 45:747–757.
- Sicardi, E., Fort, H., Vainstein, M., and Arenzon, J. (2009). Random mobility and spatial structure often enhance cooperation. *Journal of Theoretical Biology*, 256:240–246.
- Vainstein, M., Silva, A. T., and Arenzon, J. (2006). Does mobility decrease cooperation? *Journal of Theoretical Biology*, 244:722–728.
- Janssen, M. and Goldstone, R. (2006). Dynamic-persistence of cooperation in public good games when group size is dynamic. *Journal of Theoretical Biology*, 234:134–142.
- Killingback, T., Bieri, J., and Flatt, T. (2006). Evolution in groupstructured populations can resolve the tragedy of the commons. *Proceedings of the Royal Society B: Biological Sciences*, 273:1477–1481.
- Ichinose, G. and Arita, T. (2007). Cooperation achieved by migration and evolution in a multilevel selection context. *Proceedings of the 2007 IEEE Symposium on Artificial Life*, pages 236–242.
- Ichinose, G. and Arita, T. (2008). The role of migration and founder effect for the evolution of cooperation in a multilevel selection context. *Ecological Modelling*, 210:221–230.
- Pepper, J.W. and Smuts, B. B. (2002). A mechanism for the evolution of altruism among non-kin: positive assortment through environmental feedback. *American Naturalist*, 160:205–212.

Pepper, J. W. (2007). Simple models of assortment through environmental feedback. *Artificial Life*, 13:1–9.

Suzuki, S. and Kimura, H. (2011). Oscillatory dynamics in the coevolution of cooperation and mobility. *Journal of Theoretical Biology*, 287:42–47.

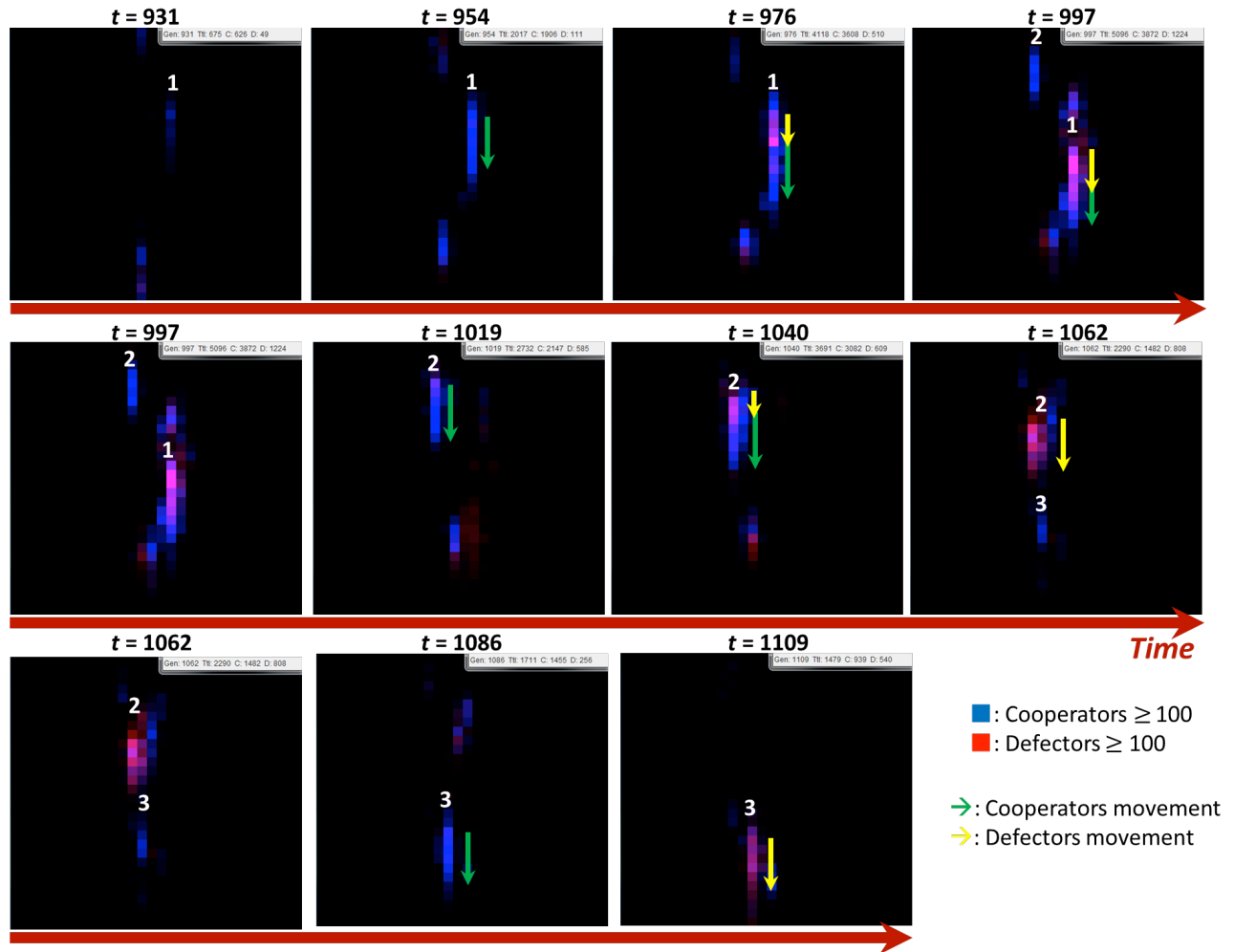


Figure 1: Typical snapshots of the downward evolution. Blue represents cooperators and red represents defectors. If the number of each type is less than 100, each color (blue or red) is reduced gradually. If there are two types of individuals in the same site, the colors are mixed. Green lines indicate the movement of cooperators and yellow lines indicate the movement of defectors. The numbers next to each group are labels, and red lines with arrows indicate the flow of time. The box located at the top-right space in each square indicates generation, number of cooperators, and number of defectors. After cooperative clusters emerged and moved downward, they were chased and exploited by defectors. Finally, each population collapsed.

On Symbiotic Policy Search and Multi-level Selection

Stephen Kelly, Peter Lichodziejewski and Malcolm I. Heywood

Faculty of Computer Science, Dalhousie University, NS. B3H 1W5
 {skelly,piotr,mheywood}@cs.dal.ca

Extended Abstract

Constructing policies for temporal sequence learning is framed as an explicitly hierarchical process of symbiosis. Two independent cycles of evolution are conducted. The symbiotic formulation assumes each policy takes the form of a cooperative team between multiple symbiont programs. A first cycle of evolution results in policies that lack the capability to generalize to solving the entire task. However, diversity is enforced to encourage the population to cover a wide range of potentially useful policies. The second cycle of evolution repeats the process but now the policies developed during the first cycle of evolution become the actions available to symbiont programs. The performance of multi-level policies is shown to be significantly better than the sum of contributing policies.

Symbiosis promotes complexification and the division of labour through (egalitarian) ecological fusion as opposed to (fraternal) reproductive fission (Queller 2000). In distinguishing explicitly between a higher-level (host) and lower level (symbiont) entities we also recognize that selection is now a multi-level concept. (Okasha 2006) distinguishes between two forms for multi-level selection (MLS) that are applicable to symbiotic models of inheritance. MLS1 defines fitness of a host as the average fitness of the symbiont membership. Conversely, MLS2 measures fitness as that defined by the host behaviour alone. Okasha goes on to make the case for assuming MLS1 during a developmental phase prior to the appearance of symbiotic (group) relationships, but adopts MLS2 once hosts (cf., groups) exist. In this work we explicitly adopt MLS2 from the outset as our interest lies in evolving hierarchies of programs for increasingly abstract decision making under temporal sequence learning tasks cf., reinforcement learning.

The generic architecture for the Symbiotic Bid-Based (SBB) policy search algorithm explicitly enforces symbiosis by separating host and symbiont into independent populations (single level of Figure 1). Each host represents a candidate solution in the form of a subset of symbionts existing independently in the symbiont population. Performance is measured relative to the interaction between a subset of initializations from the task domain (content of the point population) and host. See (Lichodziejewski 2010) for the basic SBB architecture. Extending non-hierarchical SBB to the case of hierarchical policy search is achieved by repeatedly calling the non-hierarchical scheme (multiple levels of Figure 1). Thus, a cycle of evolution begins at *level-0* for a fixed number of generations. Symbionts in level-0 are limited to the atomic actions defined by the task domain. During the next cycle of evolution (*level-1*) the process repeats under a new point-host-symbiont partnership with the level-0 host-symbionts frozen. This time symbionts assume actions defined by previously evolved hosts cf., policies identified during the level-0 cycle. Evolution is therefore bottom-up, but evaluation of any single policy is top down. For further details see (Lichodziejewski 2011; Lichodziejewski et al. 2011; Kelly et al. 2012; Doucette et al. 2012).

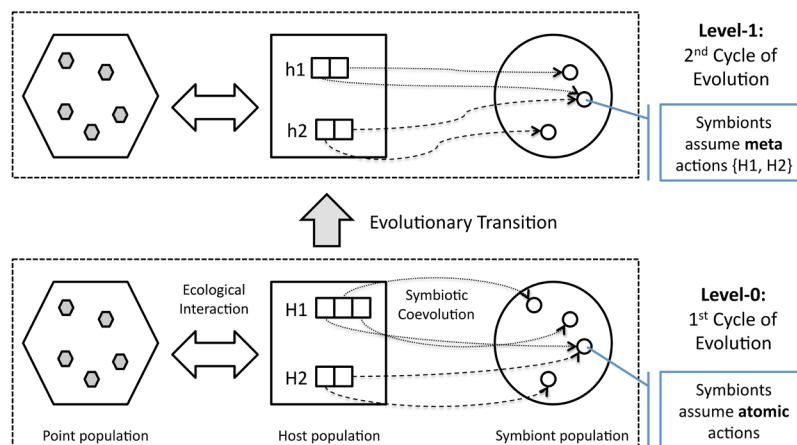


Figure 1: Generic architecture of Hierarchical Symbiotic Bid-Based GP (SBB).

Experiments are conducted under the Truck Reversal domain, a complex, non-linear control task in which an agent must back a semi-trailor up to a loading dock while avoiding a wall in the centre of the space. Policies must therefore be capable of goal seeking and obstacle avoidance (Lichodziejewski 2011, Lichodziejewski et al. 2011). During evolution, the point population specify starting configurations of the truck; whereas the host-symbionts assume responsibility for the steering behaviour.

Two experiments are considered in this work: **Case 1 – Single level**; SBB only builds a single level, thus no capacity exists for defining hierarchical policies. **Case 2 – Two level**; This scenario introduces hierarchical policy discovery (two levels).

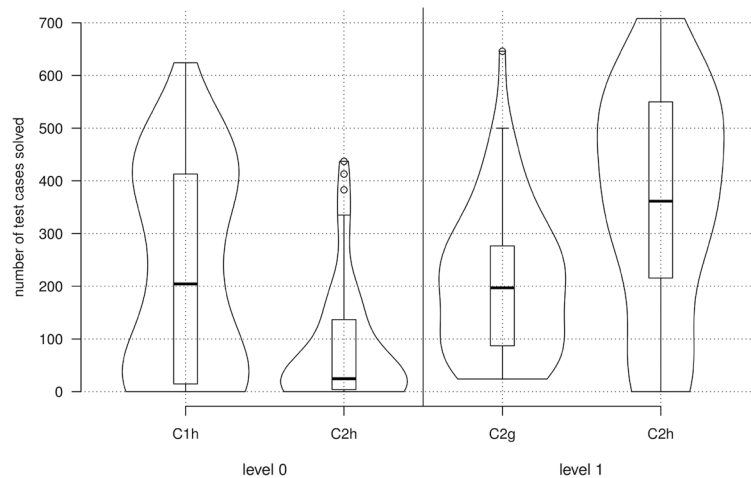


Figure 2: Generalization performance or count of number of test points solved (y-axis). Cx distinguishes between different SBB deployments or Cases 1 and 2.

Post training evaluation is conducted relative to 1000 unique start positions for the truck. Figure 2 summarizes the performance of each experiment in terms of the number of test cases solved per level. Each violin plot depicts the distribution of results over 60 independent trials. Distributions denoted 'Cxh' represent the number of test cases solved by the single best / champion host as identified w.r.t a separate validation set. For example the champion individuals in our non-hierarchical experiment, or Case 1, are able to solve roughly 200 test cases (C1h, Figure 2). Case 2 introduces hierarchical policy discovery. Champion hosts at level 0 are now typically only capable of solving 25 cases (C2h, level 0, Figure 2). However, the hosts making up the final population from level 0 now become the subset of policies available as actions to level-1 symbionts. These level-1 symbionts evolve contexts in which to deploy level 0 host-symbionts. The resulting champion individuals at level 1 (C2h, level 1, Figure 2) have succeeded in leveraging the host-symbiont policies provided at level 0. Furthermore, the performance of a level-1 host is not simply a sum over the performance of the level-0 hosts it indexes. Distribution 'C2g' (Figure 2) depicts the cumulative number of unique test cases solved by all level-0 hosts from level-1 champion hosts. Clearly, the generalization ability of a level-1 individual (MLS2) is greater than all its team members combined (MLS1). We see this as a critical requirement for characterizing a successful 'evolutionary transition'.

References

- Doucette, J., Lichodziejewski, P., and Heywood, M. (2012). Hierarchical task decomposition through symbiosis in reinforcement learning. In ACM GECCO.
- Kelly, S., Lichodziejewski, P., and Heywood, M. I. (2012). On run time libraries and hierarchical symbiosis. In IEEE CEC.
- Lichodziejewski, P. (2011). A symbiotic bid-based framework for problem decomposition using Genetic Programming. PhD thesis, Faculty of Computer Science, Dalhousie University. <http://web.cs.dal.ca/~mheywood/Thesis/PhD.html>
- Lichodziejewski, P., Doucette, J. A., and Heywood, M. I. (2011). A symbiotic framework for hierarchical policy search. Technical Report CS-2011-06, FCS, Dalhousie University. <http://www.cs.dal.ca/research/techreports/cs-2011-06>
- Lichodziejewski, P. and Heywood, M. I. (2010). Symbiosis, complexification and simplicity under GP. In ACM GECCO.
- Okasha, S. (2005). Multilevel selection and the major transitions in evolution. *Philosophy of Science*, 72:1013–1025.
- Queller, D. C. (2000). Relatedness and the fraternal major transitions. *Philosophical Transactions of the Royal Society of London: Series B*, 355:1647–1655.

Limitations of response thresholds models of division of labor

Paweł Lichocki¹, Danesh Tarapore^{1,2}, Laurent Keller² and Dario Floreano¹

¹Laboratory of Intelligent Systems (LIS), Ecole Polytechnique Fédérale de Lausanne (EPFL), Station 11, 1015, Lausanne, Switzerland

²Department of Ecology and Evolution (DEE), University of Lausanne (UNIL), Dorigny, 1015, Lausanne, Switzerland
pawel.lichocki@epfl.ch

Extended Abstract

High levels of cooperation are often cited as the primary reasons for the ecological success of social insects (Oster and Wilson, 1978; Hölldobler and Wilson, 1990). In social insects, workers perform a multitude of tasks such as foraging, nest construction and brood rearing without central control of how work is allocated among individuals (Gordon, 1996). It has been suggested that workers choose a task by responding to stimuli gathered from the environment (Robinson, 1992). Response threshold models assume that individuals in a colony vary in the stimulus intensity (response threshold) at which they begin to perform the corresponding task (see Beshers and Fewell (2001)). In (Lichocki et al., 2012), we investigated the limitations of the models of division of labor that base on the response thresholds. This abstract is meant to convey a brief summary of the points we raised in that study.

The two most often used models of division of labor are the deterministic response threshold model (DTM; Page Jr and Mitchell (1998)), and the probabilistic response threshold model (PTM; Bonabeau et al. (1996)). Both models assume that all workers receive information of the colony needs via commonly perceived stimuli. With the DTM each worker performs the task with the highest positive difference between the stimulus and its own corresponding response threshold. If all the stimuli are lower than the corresponding thresholds the worker remains idle. With the PTM the relation between stimulus and threshold is interpreted as a probability to perform the task. While these response threshold models are frequently used to explain division of labor in colonies of social insects (Bertram et al., 2003; Graham et al., 2006; Jeanson et al., 2007), no attempts have been made to quantify their efficiency in task allocation. In (Lichocki et al., 2012), we showed with formal analysis and quantitative simulations that DTM (Page Jr and Mitchell, 1998) and PTM (Bonabeau et al., 1996) lead to sub-optimal colony performance under some stimulus conditions. To overcome these problems we proposed an extended response threshold model (ETM) that can result in an efficient task allocation for any stimulus conditions. We experimentally compared all models by means of directed

evolution (see, e.g., Floreano and Keller (2010)) in a foraging scenario that required a dynamic re-allocation of workers to different tasks according to colony needs (Tarapore et al., 2010).

The common understanding of the response threshold models is that the workers' tendency to perform various tasks depends on its thresholds and that, by changing the threshold values, the worker can express any behavior, from generalist (switching between tasks) to specialist (dedicated to a specific task) (Robinson, 1992; Bonabeau et al., 1996; Beshers and Fewell, 2001). However, a mathematical analysis of the DTM reveals that the worker's behavioral flexibility depends not only on the worker's thresholds, but also on the difference between stimulus intensities. In particular, a worker can switch from task A to task B, only if there is a decrease in the difference between stimulus intensities of task A and task B. A worker can switch back from task B to task A, only if there is an increase of the aforementioned difference. Thus, contrary to the intuition standing behind the response threshold models (Robinson, 1992), the workers' behaviors are influenced not only by the absolute intensities of the stimuli, but also by their relative intensities. Consequently, the values of the stimuli constrain the worker's ability to switch tasks regardless of the values of the individual thresholds. In the PTM this constraint is less marked, because the workers' responses are stochastic, thus allowing them to switch tasks more easily. However, stochastic individual responses make the response at the colony level more unreliable, even under fixed stimuli conditions (i.e., for the same stimuli intensities the response of a worker may be different, due to its random component). Thus, both the DTM and the PTM have limitations, which could be detrimental to colony performance (Fig. 1). These problems can be overcome by extending the DTM with additional variables that weigh stimuli (ETM). The weights relax the constraints on the flexibility of task allocation by allowing the workers to scale the stimuli if needed. At the same time, the deterministic decision rules employed in the ETM allow the workers to precisely respond to changing colony needs.

Overall, our analyses highlighted the limitations of the re-

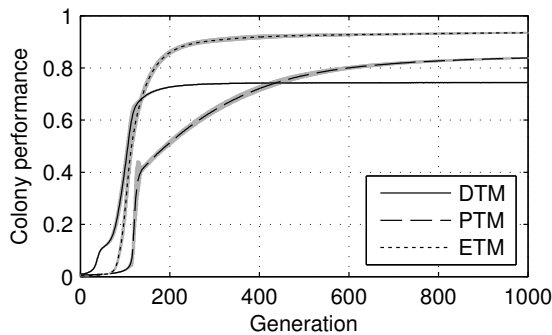


Figure 1: Mean \pm s.d. (in grey) performance with the deterministic (DTM), probabilistic (PTM), and extended (ETM) response threshold models over 1000 colonies (30 replicates). To quantify the workers' performance in task allocation we used a stochastic agent-based simulation to model a situation in which workers had to perform two distinct tasks. Our aim was to mimic situations with two vital tasks such as foraging and regulation of nest temperature. If the colony is efficient in foraging but does not regulate nest temperature well, the brood may die. Conversely, if nest temperature is well regulated, but little food is collected, only few offspring can be reared. Thus, the performance was high only if the workers efficiently performed both the regulatory and foraging tasks.

sponse threshold models that are currently used in the literature (see, e.g., Bonabeau et al. (1996); Page Jr and Mitchell (1998); Bertram et al. (2003); Graham et al. (2006); Jeanson et al. (2007)). We extended these models by weighting the stimuli. In (Lichocki et al., 2012), we also showed that the response threshold models can be formulated as artificial neural networks (see, e.g., (Haykin, 1998)). Artificial neural networks have been successfully used to control the behaviour of individuals in a colony (see e.g. Floreano et al. (2007); Waibel et al. (2009)) making it a useful approach to consider in modeling task allocation in social insects. The neuronal formalism will be useful for further extension of models, e.g., changing the threshold values with age or the integration of adaptive learning. Consequently, it constitutes a comprehensive framework for modeling task allocation in social insects. Finally, it is worth mentioning that although threshold models have been developed to explain division of labor in social insect, they may also be used to devise efficient systems of task allocation and dynamic scheduling in engineering (see, e.g., Campos et al. (2000); Bonabeau et al. (2000)).

Acknowledgments

This work was supported by the Swiss National Science Foundation and an ERC advanced grant.

References

- Bertram, S., Gorelick, R., and Fewell, J. (2003). Colony response to graded resource changes: an analytical model of the influence of genotype, environment, and dominance. *Theor Popul Biol*, 64:151–162.
- Beshers, S. and Fewell, J. (2001). Models of division of labor in social insects. *Annu Rev Entomol*, 46:413–440.
- Bonabeau, E., Dorigo, M., and Theraulaz, G. (2000). Inspiration for optimization from social insect behaviour. *Nature*, 406(6791):39–42.
- Bonabeau, E., Theraulaz, G., and Deneubourg, J. (1996). Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies. *Proc R Soc B*, 263:1565–1569.
- Campos, M., Bonabeau, E., Theraulaz, G., and Deneubourg, J. (2000). Dynamic scheduling and division of labor in social insects. *Adaptive Behavior*, 8(2):83–92.
- Floreano, D. and Keller, L. (2010). Evolution of adaptive behaviour in robots by means of darwinian selection. *PLoS Biol*, 8(1):e1000292.
- Floreano, D., Mitri, S., Magnenat, S., and Keller, L. (2007). Evolutionary conditions for the emergence of communication in robots. *Curr Biol*, 17:514–519.
- Gordon, D. (1996). The organization of work in social insect colonies. *Nature*, 380(6570):121–124.
- Graham, S., Myerscough, M., Jones, J., and Oldroyd, B. (2006). Modelling the role of intracolony genetic diversity on regulation of brood temperature in honey bee (*Apis mellifera* L.) colonies. *Insectes Soc*, 53:226–232.
- Haykin, S. (1998). *Neural networks: a comprehensive foundation*. Prentice Hall, Upper Saddle River, 2nd edn edition.
- Hölldobler, B. and Wilson, E. (1990). *The ants*. Belknap Press, Cambridge.
- Jeanson, R., Fewell, J., Gorelick, R., and Bertram, S. (2007). Emergence of increased division of labor as a function of group size. *Behav Ecol Sociobiol*, 62:289–298.
- Lichocki, P., Tarapore, D., Keller, L., and Floreano, D. (2012). Neural networks as mechanisms to regulate division of labor. *The American Naturalist*, 179(3):391–400.
- Oster, G. and Wilson, E. (1978). *Caste and ecology in the social insects*. Princeton University Press, Princeton.
- Page Jr, R. and Mitchell, S. (1998). Self-organization and the evolution of division of labor. *Apidologie*, 29:171–190.
- Robinson, G. (1992). Regulation of division of labor in insect societies. *Ann Rev Entomol*, 37:637–665.
- Tarapore, D., Floreano, D., and Keller, L. (2010). Task-dependent influence of genetic architecture and mating frequency on division of labour in social insect societies. *Behav Ecol Sociobiol*, 64(4):675–684.
- Waibel, M., Keller, L., and Floreano, D. (2009). Genetic team composition and level of selection in the evolution of multi-agent systems. *IEEE Trans Evol Comput*, 13:648–660.

A Synthetic Ecology Model for the Educational Game *Calangos*

Angelo Loula^{1*}, Leandro N. de Castro², Antônio L. Apolinário Jr.³,
Pedro L. B. da Rocha⁴, Charbel N. El-Hani⁵

¹ Intelligent and Cognitive Systems Lab, State University of Feira de Santana, Brazil

² Natural Computing Laboratory, Mackenzie Presbyterian University, Brazil

³ Computer Science Department, Federal University of Bahia, Brazil

⁴ Laboratory of Terrestrial Vertebrates, Institute of Biology, Federal University of Bahia, Brazil

⁵ History, Philosophy and Biology Teaching Laboratory, Institute of Biology, Federal University of Bahia, Brazil

*corresponding author: angelocl@ecomp.uefs.br

Extended Abstract

Ecology deserves special attention in biological education due to the fact that its object, namely, the spatial and temporal patterns of distribution and abundance of organisms, as well their causes and consequences, plays a central role in biology (Scheiner & Willig, 2008; Scheiner, 2010). We have developed an electronic game as a resource aiming at promoting students' learning about ecology, by making ecological concepts more concrete to the students and, also, engaging them with conceptual learning in ecology in a more active manner.

The game, called *Calangos* (freely available at <http://calangos.sourceforge.net/>), is based on a real ecological case situated in the dunes of the middle São Francisco River, in the state of Bahia, Brazil, investigated by researchers from Brazil and abroad (e.g. Rocha et al., 2004). The game is intended to provide the students with an environment showing sufficient realism, so as to allow an adequate understanding of ecological processes. An important step in game development was the model of the synthetic ecological system, based on the real ecological case, included in the game.

Calangos is a simulation and action game with 3D visualization in first and third person. The player controls a lizard from one of the three medium-sized endemic species (*Tropidurus psammonastes*, *Cnemidophorus sp. nov.*, and *Eurolophosaurus divaricatus*). The player begins as a lizard in the start of its life, situated in the dunes terrain, in which there are relevant elements from the ecosystem that can be involved in ecological relations with the player-controlled lizard. It is expected that the student makes use of concepts related to different ecological relationships in order to overcome the challenges faced by the lizard to survive, develop and reproduce successfully.



Figure 1: Left: A predator (seriema) attacking the lizard. Right: Male and Female lizards close to the player's lizard.

In order to build a synthetic ecosystem for this computer game, based on the real ecological case of the São Francisco River dunes, we relied on the literature and descriptions from ecologists concerning this region. As the player controls a lizard, the relevant ecological relationships modeled for the game were prey-lizard, predator-lizard, vegetation-lizard, lizard-lizard and lizard-physical environment. There are various species of plants, typical preys of lizards, various species of lizards' predators (figure 1). Other co-specific lizards are also present, engaging in ecological relationships (e.g. competition for territory, for preys, for breeding). Besides, there are abiotic elements that are also part of the ecosystem, such as the climate and terrain (see Loula et al., 2009). Each element and relationship was initially described by biologists that are part of the project team. The terrain, animals and vegetation were visually modeled in three dimensions, trying to reproduce their actual visual aspect. More importantly, computational models were proposed to describe all relevant elements and relations and establish the game simulation dynamics.

The computational model developed defines a complex network of interrelated elements. To achieve survival and reproductive success the player must define its strategy, such as when and what to hunt and eat. But to better define a game strategy, the player must understand the game mechanics and, therefore, must comprehend the ecological dynamics.

References

- Loula, A.C., Oliveira, E.S.de, Munoz, Y. J., Vargens, M. M. F., Apolinario Jr., A. L., Castro, L.N., Rocha, P., El-Hani, C.N. (2009). Modelagem Ambiental em um Jogo Eletrônico Educativo. In *VIII Brazilian Symposium on Games and Digital Entertainment*, SBGames 2009, Rio de Janeiro.
- Rocha, P.L.B.da, Queiroz, L.P.de, Pirani, J.R. (2004) Plant species and habitat structure in a sand dune field in the brazilian Caatinga: a homogeneous habitat harbouring an endemic biota. *Revista Brasileira de Botânica*, 27(4):739-755.
- Scheiner, S. M. & Willig, M. R. (2008). A general theory of ecology. *Theoretical Ecology* 1:21-28.
- Scheiner, S. M. (2010). Toward a conceptual framework for biology. *The Quarterly Review of Biology* 85(3): 293-318.

The Role of Memory in Stabilizing Swarms

Jennifer M. Miller¹, Hao Luan², Louis F. Rossi¹, Chien-Chung Shen²

¹Department of Mathematical Sciences, University of Delaware, Newark DE 19716

²Department of Computer and Information Sciences, University of Delaware, Newark DE 19716
rossi@math.udel.edu

Extended Abstract

Simulations of self-organized swarms frequently use discrete time steps to approximate swarm dynamics. In order to demonstrate the effect of memory on swarm dynamics, we analyze and evaluate swarm interactions using varying amounts of kinetic memory. We define kinetic memory as the stored velocity states for n discrete time steps in the past. It is reasonable to suppose that individuals in a swarm possess a memory of the immediate past and use this information to their advantage when swarming. We show that kinetic memory can play a key role in the dynamics of biological and artificial aggregations.

Individual-based models (IBMs) of biological aggregations generally calculate an individual's new state based on the current state and a fixed time step, e.g. (Couzin et al., 2002; Giardina, 2008; Huth and Wissel, 1992; Cucker and Smale, 2007). Some simulations of biological aggregations attempt to use this discrete time step to represent the physiological reaction time of the species being modeled, but there has been little effort to investigate the role of memory in this special context. In the broader context, memory is a central theme in most computational frameworks and has been used to expand or augment dynamic systems. For instance, Ho et. al. proposed a conceptual framework to store episodic memory to make agents more adaptable (Ho et al., 2008). Similarly, Mirza et. al. designed a scheme for storing of sensorimotor experiences and interactions to make robots more adaptable (Mirza et al., 2007). From a more algorithmic perspective, others have explored the use of stored memory of past states in particle swarm optimization problems (see (Wang and Wang, 2007)), but the effect of memory on swarm dynamics has received very little attention. Because the storage of past movements requires no communication in robotics applications, memory can be used to stabilize aggregations. In fact, the communication rate between nearby individuals in wireless robotics networks is more limited than in many biological applications, so the time step used to update an individual's state is necessarily greater. To swarm effectively, an individual must gather information about the relative position and speed of nearby individuals. However, the rate at which this information can

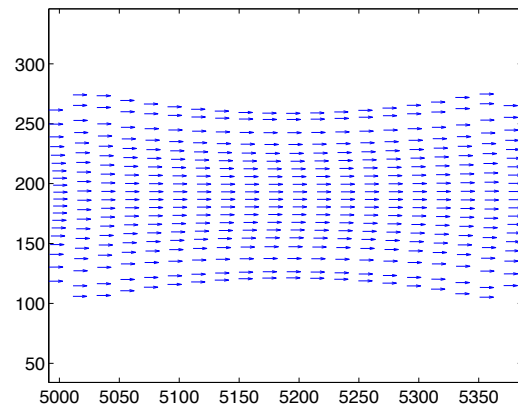


Figure 1: A non-axisymmetric equilibrium swarm configuration.

be gathered using a wireless network is limited by factors such as channel bandwidth, packet collisions and other non-ideal effects.

In order to explore the role of memory in swarms, we investigate the connections between numerical methods for ordinary differential equations and swarm dynamics. Specifically, we explore two stable configurations that arise for a particular set of parameters. Either state is possible. One is non-axisymmetric (see Figure 1 for a discrete representation) and the other is axisymmetric (see Figure 2). Both solutions to the continuum system are stable, but when approximated in discrete time, they can be unstable depending upon the discretization and the time step. We can approximate the continuum model of swarming by an individual-based model, effectively converting a system of partial differential equations to a system of ordinary differential equations, which is then discretized in time and integrated forward. The system can be integrated more accurately by using previous individual states as well as the current state. For example, the Adams-Bashforth schemes are explicit multistep meth-

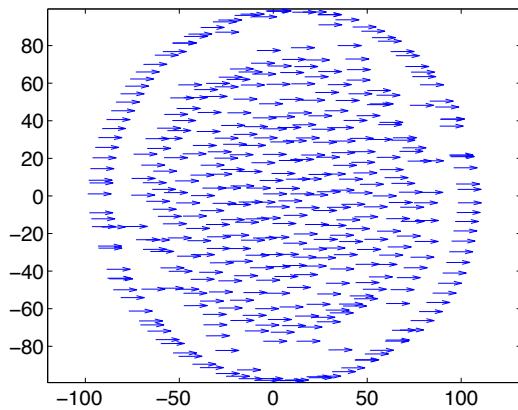


Figure 2: An axisymmetric equilibrium swarm configuration.

ods that incorporate stored derivatives from the n previous states in determining the next state. Because we need to minimize communication, we do not consider implicit schemes like Adams-Moulton, which provide greater numerical stability. Implicit schemes require more communication, which is a prohibitive drawback in wireless networks.

We have also conducted wireless network simulations using the QualNet simulator and compared the results to simulations with ideal communications in order to see if kinetic memory can be used to remediate the communication bottleneck. In the QualNet simulations, robots are wirelessly connected via the IEEE 802.11 protocol for information transfer. The robots broadcast their location and velocity information for each time step to the neighboring robots. Because broadcasts can collide, each time step is further divided into multiple time slots, each of which is then assigned to a different robot exclusively. Each robot can only broadcast in its own time slot, and so collisions are avoided. However, wireless communications have some constraints. First of all, broadcasts can only reach a limited range. If robots are sparsely distributed, they may only be able to exchange information with a small number of other individuals. Next, the time step size has a lower bound. Due to the finite bandwidth, there is a minimum time required for each broadcast, and so each time slot must be big enough to hold an individual's complete broadcast. If the time step length is below the lower bound, packet collisions may occur. Finally, wireless communications in real scenarios suffer from channel fading, interference, and other environmental impairments.

Using a three-zone continuum swarming model (Miller et al., 2012), we have determined the eigenvalues of the dynamical system corresponding to a stable, anisotropic, translating aggregation. These eigenvalues allow us to calculate the largest time step for which a numerical scheme is sta-

ble. In other words, we can predict the stability threshold for IBMs with different amounts of kinetic memory. We compare the stability thresholds of an Euler (single step) scheme and a multistep scheme. We find that a predictor-corrector method with a tunable parameter provides numerical stability even with long time steps. Adjusting the parameter allows us to change the shape and size of the region of absolute stability for the numerical method. In our case, the eigenvalues are clustered close to the negative real axis and we have tuned the parameter to extend the region as far as possible along this axis. We find that QualNet simulations of swarms of wireless robots are aligned with our theoretical predictions. The threshold of the time step is similar, and the groups exhibit the same dynamics. As a result, we see that kinetic memory offers an distinct advantage in both biological and artificial swarms.

References

- Couzin, I., Krause, J., James, R., Ruxton, G., and Franks, N. (2002). Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, 218:1–11.
- Cucker, F. and Smale, S. (2007). Emergent Behavior in Flocks. *IEEE Transactions on Automatic Control*, 52(5):852–862.
- Giardina, I. (2008). Collective behavior in animal groups : theoretical models and empirical studies. *HFSP Journal*, 2(4):205–219.
- Ho, W. C., Dautenhahn, K., and Nehaniv, C. (2008). Computational memory architectures for autobiographic agents interacting in a complex virtual environment: a working model. *Connection Science*, 20(1):21–65.
- Huth, A. and Wissel, C. (1992). The simulation of the movement of fish schools. *J. theor. Biol.*, 156:365–385.
- Miller, J. M., Kolpas, A., Juchem Neto, J. P., and Rossi, L. F. (2012). A continuum three-zone model for swarms. *Bulletin of Mathematical Biology*, 74(3):536–61.
- Mirza, N. a., Nehaniv, C. L., Dautenhahn, K., and te Boekhorst, R. (2007). Grounded Sensorimotor Interaction Histories in an Information Theoretic Metric Space for Robot Ontogeny. *Adaptive Behavior*, 15(2):167–187.
- Wang, H. and Wang, D. (2007). Triggered memory-based swarm optimization in dynamic environments. *Applications of Evolutionary Computing*, pages 637–646.

A Bottom-Up Approach to the Evolution of Swarming

Randal S. Olson^{1,3,5}, Christoph Adami^{2,5}, Fred C. Dyer^{3,4,5} and Arend Hintze^{1,2,5}

¹*Department of Computer Science and Engineering*, ²*Department of Microbiology and Molecular Genetics*,

³*Ecology, Evolutionary Biology, and Behavior Program*, ⁴*Department of Zoology*,

⁵*BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, MI 48823, USA*

Extended Abstract

One of nature's most evident examples of self-organization is the formation of swarms, schools, or flocks of animals. These groups of individuals coordinate their movement on an individual basis to form self-organized collectives. It has been hypothesized that these aggregations of individuals improve mating success (Diabate et al., 2011), or may be an adapted defense against predators by confusing the potential predator (Krause and Ruxton, 2002; Jeschke and Tollrian, 2007). In the past, these ostensibly complex swarming behaviors have been explained by the swarm members adhering to three simple rules: 1) Move in the same direction as your neighbors; 2) Remain close to your neighbors; and 3) Avoid collisions with your neighbors (Reynolds, 1987). We characterize this model as a top-down approach, where the behavior of the group is explained by simple rules that were conceived ad hoc and work only when applied to that particular system. Generally, this approach requires knowledge about the position and motion vector of nearby agents and therefore requires complex mathematical computations to determine the motion of each agent in the swarm (Oboshi et al., 2002; Chen and Fang, 2006; Hemelrijk and Hildenbrandt, 2011).

We find it implausible that biological creatures in swarms are performing complex computations, such as determining the relative position and motion vector of nearby conspecifics, every millisecond to make a decision about where to move next. We suggest instead that there must be a simpler, more computationally tractable mechanism (for biological organisms) that is guiding swarming behavior in nature. In this abstract we present a bottom-up approach, where each agent in the swarm is controlled individually by a Markov network brain (Edlund et al., 2011) as opposed to genetic programming (Reynolds, 1993) or neural networks (Kwasnicka et al., 2007). The information provided to each swarm agent is limited to the information that the agent's retina conveys, and every agent's actions depend only on a combination of the swarm agent's current sensory input (e.g., eyes and ears) and the state of internal nodes in the swarm agent's Markov brain (i.e., memory). We suggest that this is a more realistic model of swarms observed in

nature, since the information provided to the brain is simple to compute and decisions are made on an individual basis rather than by a top-down controller. This evolutionary agent-centered approach enables us to examine the environmental conditions that are conducive for swarming, and how these conditions influence the evolution of swarming behavior.

In nature, we observe two varieties of swarming behavior: insect swarms which remain at one location during their breeding period to facilitate mating (Diabate et al., 2011), and flocks of birds or schools of fish that roam while still maintaining a coherent swarm. Swarm coherence is believed to be influenced by the rate of predation (Beauchamp, 2004), thus some swarming behaviors can be understood as a group effort to deter potential predators (Krause and Ruxton, 2002; Jeschke and Tollrian, 2007). Examples of anti-predator swarming behavior can be observed in nature, such as in flocks of starlings (Feare, 1984). While predation is believed to be the key selection pressure causing the difference between stationary and roaming swarms, there is little evidence to support this (Beauchamp, 2004). Evolutionary experiments on natural swarms are inconvenient and time-consuming, while our bottom-up approach of evolving agent controllers allows these questions to be addressed in an experimental model system.

Every swarm agent has its own retina consisting of two rows of 12 pixels covering a range of 180° facing forward. Each of the 12 pixels covers a 15° segment and indicates if at least one other swarm agent is within viewing range within that segment. The second row of pixels functions identically to the first, but instead indicates the presence of a predator. Each swarm agent is controlled by its own Markov network brain, defined by a network of Markov variables that are connected by stochastic logic gates (as in Edlund et al. 2011), except that we also allow deterministic along with stochastic gates. We evolve the Markov network brains with a standard Genetic Algorithm, where mutations alter the brain by adding or removing connections between input, output, and memory nodes, or modifying the logic of one of the brain's Markov gates. The swarm agents have the

choice every update to travel straight ahead at a speed of 1 unit (normal speed), to travel straight ahead at a speed of 2 units (rushing speed), or to travel a distance of 1 unit and turn left or right by 8° (turning), for a total of 4 possible actions. In the experiments where we study the effects of predation on swarm behavior, we include a hand-designed predator that performs swooping attacks on the swarm. The predator has a retina covering 40° in front of it and targets agents in its field of view with a probability of $\frac{1}{d}$ where d is the agent's distance from the predator, such that closer agents are more likely to be targeted for predation. In simulation, the predator moves at a constant speed of 1.5 units and has a 25% chance of successfully killing any swarm agent that gets within 3 units of it.

We used three different fitness functions to evolve the swarms: rewarding coherence, rewarding avoidance of the predator, and rewarding avoidance of the predator while also maintaining coherence. The fitness of a swarm being rewarded for coherence is $W_s = \sum_{t=0}^{t_{\max}} \sum_{i=1}^n \frac{1}{r_i}$, where n is the number of agents alive in the swarm (here, $n = 20$), t_{\max} is the total number of updates for which the swarm is evaluated, and r is the distance of the agent to the center of the swarm at update t . The fitness of a swarm under predation is computed as $W_p = \sum_{t=0}^{t_{\max}} \sum_{i=1}^n d_i$, where d is defined as the distance between agent and predator. Dead agents have a distance of 0 to the predator. If both selection pressures for coherence and predation are applied, the total fitness is the sum of both components: $W_s + \frac{1}{4}W_p$. Each of the three selection regimes were tested in 100 replicate experiments with a standard Genetic Algorithm with fitness proportional selection, 1% per-gene mutation rate, 5% gene duplication, and 2% deletion rate, and no cross-over.

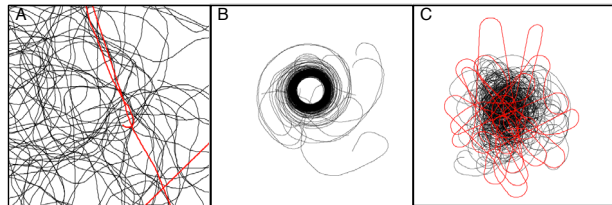


Figure 1: Trajectories of individuals in swarms with only predation (A), with only rewarding coherence (B), and with predation and rewarding coherence (C). Swarming agent paths in black, predator paths in red. All three figures have the same scale.

Selecting only for predator avoidance results in complete dissipation of the swarm (Figure 1A), and shows that predation alone is insufficient for driving swarming behavior in this system. On the other hand, selecting for coherence alone results in agents that aggregate but move in small, predictable circles that do not roam (Figure 1B). When selecting for predation avoidance and coherence at the same time, some swarms show similar behavior than those evolved

without predation, but we also find several swarms that actively avoid the predator and roam unpredictably (Figure 1C), similar to preyed swarms observed in nature. Taken together, these results demonstrate that realistic swarming behavior can be evolved in an agent-based model with minimal information provided to each agent, suggesting that more complex models (e.g., models that require processing of relative positions and motion vectors) are not proper models of natural swarms. Our results suggest that a bottom-up approach using Markov brains represents a promising new platform that can be used to study the evolution of swarming behaviors in an experimental system.

References

- Beauchamp, G. (2004). Reduced flocking by birds on islands with relaxed predation. *Proceedings of the Royal Society London Series B*, 271(1543):1039–1042.
- Chen, S. and Fang, H. (2006). Modeling and control of scalable engineering swarm. In *The Sixth World Congress on Intelligent Control and Automation*, pages 526–530, Piscataway, N.J. IEEE Press.
- Diabate, A., Yaro, A., Dao, A., Diallo, M., Huestis, D., and Lehmann, T. (2011). Spatial distribution and male mating success of *Anopheles gambiae* swarms. *BMC Evolutionary Biology*, 11(1):184.
- Edlund, J. A., Chaumont, N., Hintze, A., Koch, C., Tononi, G., and Adami, C. (2011). Integrated information increases with fitness in the evolution of animats. *PLoS Comput Biol*, 7(10):e1002236.
- Feare, C. (1984). *The starling*. Shire natural history. Oxford University Press, USA.
- Hemelrijk, C. K. and Hildenbrandt, H. (2011). Some causes of the variable shape of flocks of birds. *PLoS ONE*, 6(8):e22479.
- Jeschke, J. M. and Tollrian, R. (2007). Prey swarming: which predators become confused and why? *Animal Behaviour*, 74(3):387–393.
- Krause, J. and Ruxton, G. (2002). *Living in groups*. Oxford University Press, USA.
- Kwasnicka, H., Markowska-Kaczmar, U., and Mikosik, M. (2007). Open-ended evolution in flocking behaviour simulation. In *Proceedings of the International Multiconference on Computer Science and Information Technology*, pages 103–120.
- Oboshi, T., Kato, S., Mutoh, A., and Itoh, H. (2002). Collective or scattering: evolving schooling behaviors to escape from predator. In Standish, R. K., Bedau, M. A., and Abbass, H. A., editors, *Proceedings of the 8th International Conference on Artificial Life*, pages 386–389, Cambridge, MA, USA. MIT Press.
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '87, pages 25–34, New York, NY, USA. ACM.
- Reynolds, C. W. (1993). An evolved, vision-based behavioral model of coordinated group motion. In *From animals to animats 2*, pages 384–392, Cambridge, MA, USA. MIT Press.

Can Simpson's Paradox Explain Co-operation in *Pseudomonas aeruginosa* Biofilms?

Alexandra S. Penn^{1,4}, Tim C.R. Conibear^{2,3}, Richard A. Watson¹, Alex R. Kraaijeveld² & Jeremy S. Webb²

¹Electronics and Computer Science, University of Southampton, Southampton UK

²Centre for Biological Sciences, Life Sciences Building 85, University of Southampton, Southampton UK

³Royal Free Hampstead NHS Trust, London UK

⁴Centre for Environmental Strategy, University of Surrey, Guildford UK

a.penn@surrey.ac.uk

Extended Abstract

Co-operative behaviours, such as the production of public goods, are commonly displayed by bacteria in biofilms and can enhance their ability to survive in environmental or clinical settings (Ghannoum & O'Toole 2004, Crespi 2001, West *et al.* 2007). Non-cooperative cheats commonly arise (de Vos *et al.* 2001, Schaber *et al.* 2004) and should, theoretically, disrupt co-operative behaviour (Hardin 1968, Rankin *et al.* 2007). Its stability therefore requires explanation, but no mechanisms to suppress cheating within biofilms have yet been demonstrated experimentally (e.g. Rainey & Rainey 2003, Griffin *et al.* 2004, Kreft 2004, Buckling *et al.* 2007). Theoretically, repeated aggregation into groups, interleaved with dispersal and remixing, can increase cooperation via a 'Simpson's Paradox' (Wilson 1980). That is, an increase in the global proportion of co-operators despite a decrease in within-group proportions, when frequency of cheats increases within any one group, but groups with a higher initial proportion of co-operators grow larger (Simpson 1951). Chuang *et al.* (2009) have shown that given appropriate population structure such an effect may increase the population of public-good-producing co-operators relative to non-producer cheats in a synthetic system of two strains of *Escherichia coli*. However, in that experiment any natural population structure was removed (the bacteria were maintained in a well-mixed planktonic phase throughout) and artificial population structure was imposed (by means of microtitre plate wells). Thus, the importance of Simpson's paradox in natural bacterial populations remains to be determined. Natural populations (that are neither artificially mixed nor artificially subdivided) do in fact exhibit considerable population structure consisting of distinct individual microcolonies. These colonies undergo a formation, development and dispersal process (Hall-Stoodley *et al.* 2004) which bears a striking similarity to the aggregation and dispersal process required for Simpson's paradox to maintain co-operation. Simpson's Paradox might thus explain the persistence of co-operation in the natural biofilm state.

Using the production of iron-chelating siderophores, a public good (Varma & Chincholker 2007), in *Pseudomonas aeruginosa* as our model system for co-operation, we used wild-type co-operator and siderophore-deficient cheat strains (gfp-tagged) to measure the frequency of cooperating and cheating individuals *in-situ* within living microcolony structures. The development of 17 specific microcolonies was tracked and imaged over 10 days using continuous culture in flow cells and laser confocal microscopy under conditions of iron limitation in which siderophores are necessary for cell growth (flow cell inoculation and culture is as previously described (Moller *et al.* 1998; Webb *et al.* 2003)). Microcolony and within-colony co-operator and cheat biomass was calculated from our 3D images. For full details of all methods see Penn *et al.* (2012).

We detected neither a Simpson's Paradox (global and within-colony proportions of cheats were highly and significantly correlated) nor the conditions that would be necessary for Simpson's paradox to occur: Firstly, that the proportion of cheats should always increase within microcolonies; secondly that microcolonies containing a lower proportion of cheats have an increased overall growth rate. We did however detect significant within-type negative density-dependent effects which vary over microcolony development. Microcolonies also showed characteristic changes in structure and spatial distribution of cheat and wild-type cells over their development (16 of 17 colonies developed similarly). Typically microcolonies were initially composed of wild-type cells surrounded by a few individual cheat cells. 48 hours later however, the structure of the colonies had changed distinctly to numerous cheats inside the microcolonies, surrounded by wild-type cells. This within-microcolony spatial structure coupled with limited siderophore diffusability may violate the assumption required for Simpson's paradox that group members share equally in the public good. Since Simpson's paradox is observed in Chuang *et al.*'s (2009) artificial planktonic experiment but not observed here within biofilms, assumptions about the behaviour and distribution of cheat and wild type strains that hold for theoretical and artificial conditions may not hold true in the real biofilm context. This has concomitant implications for the evolution of co-operation and its co-evolution with population structure in real biological systems.

References

- Buckling A, Harrison F, Vos M, Brockhurst MA, Gardner A, West SA & Griffin A (2007) Siderophore-mediated cooperation and virulence in *Pseudomonas aeruginosa*. *FEMS Microbial. Ecol.* **62**: 135-141
- Chuang JS, Rivoire O & Leibler S (2009) Simpson's Paradox in a Synthetic Microbial System *Science* **323**: 272
- Crespi BJ (2001) The evolution of social behaviour in microorganisms. *Trends Ecol. Evol.* **16**: 178-183
- De Vos D, De Chial M, Cochez C, Jansen S, Tummeler B, Meyer JM, Cornelis P (2001) Study of pyoverdine type and production by *Pseudomonas aeruginosa* isolated from cystic fibrosis patients. *Arch Microbiol* **175**:384-388
- Frank SA (1998) Foundations of Social Evolution. *Princeton Univ. Press*. Princeton
- Ghannoum MA & O'Toole GA eds. (2004) Microbial biofilms. *ASM Press*. Washington, DC.
- Griffin AS, West SA & Buckling A (2004) Cooperation and competition in pathogenic bacteria. *Nature* **430**: 1024-1027
- Hall-Stoodley L, Costerton JW & Stoodley P (2004) Bacterial Biofilms: From the natural environment to infectious diseases. *Nature Reviews Microbiology* **2**: 95-108
- Hardin, G (1968) The tragedy of the commons. *Science* **162**: 1243-148
- Kreft JU (2004) Biofilms promote altruism. *Micobiology* **150**: 2751-2760
- Moller S, Sternberg C, Andersen JB, Christensen BB, Ramos JL, Givskov M, Molin S (1998) In situ gene expression in mixed-culture biofilms: evidence of metabolic interactions between community members. *Appl. Environ. Microbiol.* **64**, 721-732
- Penn AS, Conibear TCR, Watson RA, Kraaijeveld AR & Webb JS (2012) Can Simpson's Paradox Explain Co-operation in *Pseudomonas aeruginosa* Biofilms? *FEMS immunology and Medical Microbiology* (in Press)
- Rainey PB & Rainey K (2003) Evolution of cooperation and conflict in experimental bacterial populations. *Nature* **425**: 72-74
- Rankin DJ, Bargum K & Kokko H (2007) The tragedy of the commons in evolutionary biology. *Trends Ecol. Evol.* **22**: 643-651
- Schaber JA, Carty NL, McDonald NA, Graham ED, Cheluvappa R, Griswold JA & Hamood A (2004) Analysis of quorum sensing-deficient clinical isolates of *Pseudomonas aeruginosa*. *Journal of Medical Microbiology* **53**: 841-853
- Simpson EH (1951) The Interpretation of Interaction in Contingency Tables. *Journal of the Royal Statistical Society B* **13**: 238-241
- Varma A & Chincholkar S eds. (2007) Microbial Siderophores, vol. 12 of *Soil Biology* Springer, Berlin/Heidelberg
- Webb JS, Thompson LS, James S, Charlton T, Tolker-Nielsen T, Koch B, Givskov M & Kjelleberg, S (2003) Cell death in *pseudomonas aeruginosa* biofilm development. *Journal of Bacteriology* **185**: 4585-4592.
- West SA, Diggle SP, Buckling A, Gardner A & Griffen AS (2007) The social lives of microbes. *Annu. Rev. Ecol. Evol. S.* **38**: 53-77
- Wilson DS (1980) The natural selection of populations and communities. *Benjamin/Cummings*, California

“Take me to your leader!” Inferring leadership in animal groups on the move

Nicolas Perony¹, Thomas O. Richardson¹, Marta B. Manser², and Frank Schweitzer¹

¹Chair of Systems Design, ETH Zurich, Zurich, Switzerland

²Institute of Evolutionary Biology and Environmental Studies, University of Zurich, Zurich, Switzerland
nperony@ethz.ch

Extended abstract

The topic of group living in animals (Krause and Ruxton, 2002), and especially the questions of how individuals share information, make collective decisions, and move as a group have been the focus of particular attention in recent scientific work (Couzin and Krause, 2003; Sumpter, 2006, 2010). An issue that is central to the problem of collective movement is how to quantitatively capture the influence of leaders, or indeed to determine whether leaders exist at all. Within a group, a leader is a key individual whose impact on the collective behaviour is significantly higher than that of other individuals (or “followers”) (King et al., 2009). Studies now abound on the emergence and the role of leaders in collective behaviour, both from an experimental (Harcourt et al., 2009; Nagy et al., 2010; Lukeman et al., 2010; Tarcai et al., 2011; Couzin et al., 2011) and theoretical perspective (Grégoire et al., 2003; Rands et al., 2003, 2008; Grégoire and Chaté, 2004; Couzin et al., 2005; Conradt et al., 2009). However, a major challenge which is posed in the study of leaders and followers in animal groups is the identification of these leaders. Indeed, when animals move along a dynamical front whose shape and direction are subject to permanent changes and fluctuations, how to reliably determine who is leading the group? Moreover, may it be that in some contexts the effective group leader is situated within the group’s core instead of at its rim, and in these cases how to capture its role? In this work we present new methods to infer and measure leadership in groups of entities moving collectively at variable speeds. We describe quantitative tools to study the synchronised trajectory of many individuals, and test these tools against simulated and measured collective movement data.

In a first step, we concentrate on the spatial dimension of leadership and ask the question of how to infer the dynamical progression order of a moving group. We present an algorithm to estimate a group’s trajectory from that of its members, and expand on how to make the estimate robust to stop-and-go motion, or the alternation between dynamical movement periods (e.g. moving between food patches) and semi-static periods (e.g. foraging at a food patch), which are less relevant for the group’s trajectory. The method is based on an estimation of the trajectory of the group’s centroid, smoothed with a low-pass filter and protected against

spurious directional changes due to noisy data.

We first apply this algorithm to the study of trajectories extracted from simulations of a multiplayer racing computer game and observe a strict correspondence between the ranking computed by the program from the progression along a known race circuit and our dynamically-computed ranking. We then use GPS tracks of human runners moving along a simple path at variable speeds and comment on the accuracy on the method. Finally, we apply our algorithm to the study of noisy trajectories coming from GPS sensors worn on a collar by individual meerkats (*Suricata suricatta* – illustrated in Fig. 1) and discuss the advantages of such a method to identify specific behaviours such as mate guarding.

In a second step, we present cases where studying leadership from a purely spatial perspective is irrelevant. This leads us to generalise our approach to leadership in moving groups by using information-theoretic measures, in particular conditional mutual information (CMI), to determine the directionality of the information flow between group members. The use of the CMI metric allows us to identify leaders and followers by inferring causality between the trajectories of individuals, thereby offering a richer definition of leadership which does not require leaders to be at the front of a moving group. Moreover, the method offers a time-dependent measure of leadership, which differs from previous work (see e.g. Nagy et al., 2010) in which aggregated measurements were averaged. We use surrogate data sets to generate synthetic uncorrelated trajectories; these provide a null model of interindividual crosscorrelation, which we use to quantify the significance of the CMI levels measured between dyads within the group. Finally, we apply this method to the set of meerkat GPS tracks used previously (illustrated in Fig. 2), and we study the influence of vocalisations (e.g. moving calls) emitted by individual animals on the flow of information between them.

These methods, and the insights obtained from them, are of relevance to the study of collective movement patterns in general. We expect that by drawing from other fields such as information theory and nonlinear time series analysis, these new tools will help to better understand the proximate factors underlying the synchronisation of behaviours between individuals within a group.

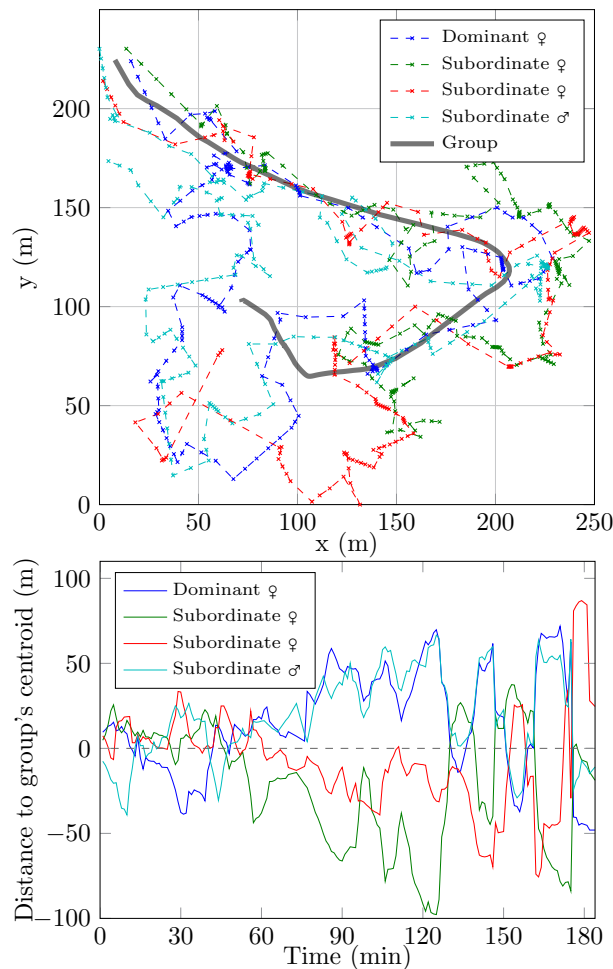


Figure 1: Application of the spatial leadership method to a group of 4 individual meerkats tracked over a period of 3 hours. (top) Thin lines: individual trajectories; thick line: group trajectory extracted from the individual trajectories. (bottom) Individual leading index, expressed as the signed distance between an individual's position projected on the group trajectory and the group's centroid.

References

- Conradt, L., Krause, J., Couzin, I., and Roper, T. (2009). Leading according to need in self-organizing groups. *The American Naturalist*, 173(3):304–312.
- Couzin, I., Ioannou, C., Demirel, G., Gross, T., Torney, C., Hartnett, A., Conradt, L., Levin, S., and Leonard, N. (2011). Uninformed individuals promote democratic consensus in animal groups. *science*, 334(6062):1578–1580.
- Couzin, I. and Krause, J. (2003). Self-organization and collective behavior in vertebrates. *Advances in the Study of Behavior*, 32:1–75.
- Couzin, I., Krause, J., Franks, N., and Levin, S. (2005). Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025):513–516.

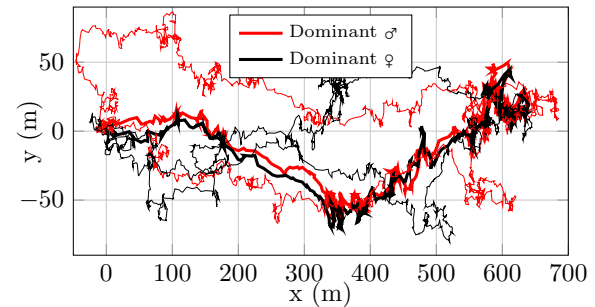


Figure 2: Trajectories of two different individuals (thick lines) followed over a period of 2h45m and an example of corresponding surrogate trajectories (thin lines – two iterates); all trajectories start at (0;0). Whilst the real trajectories appear to be crosscorrelated (as the individuals move together), the surrogates ones are not; this provides a null model to test the values of conditional mutual information obtained.

- Grégoire, G. and Chaté, H. (2004). Onset of collective and cohesive motion. *Physical Review Letters*, 92(2):25702.
- Grégoire, G., Chaté, H., and Tu, Y. (2003). Moving and staying together without a leader. *Physica D: Nonlinear Phenomena*, 181(3-4):157–170.
- Harcourt, J., Ang, T., Sweetman, G., Johnstone, R., and Manica, A. (2009). Social feedback and the emergence of leaders and followers. *Current Biology*, 19(3):248–252.
- King, A., Johnson, D., and Van Vugt, M. (2009). The origins and evolution of leadership. *Current biology*, 19(19):R911–R916.
- Krause, J. and Ruxton, G. (2002). *Living in groups*. Oxford University Press.
- Lukeman, R., Li, Y., and Edelstein-Keshet, L. (2010). Inferring individual rules from collective behavior. *Proceedings of the National Academy of Sciences*, 107(28):12576.
- Nagy, M., Ákos, Z., Biro, D., and Vicsek, T. (2010). Hierarchical group dynamics in pigeon flocks. *Nature*, 464(7290):890–893.
- Rands, S., Cowlshaw, G., Pettifor, R., Rowcliffe, J., and Johnstone, R. (2008). The emergence of leaders and followers in foraging pairs when the qualities of individuals differ. *BMC evolutionary biology*, 8(1):51.
- Rands, S., Cowlshaw, G., Pettifor, R., Rowcliffe, J., Johnstone, R., et al. (2003). Spontaneous emergence of leaders and followers in foraging pairs. *Nature*, 423(6938):432–434.
- Sumpter, D. (2006). The principles of collective animal behaviour. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 361(1465):5–22.
- Sumpter, D. (2010). *Collective animal behavior*. Princeton University Press.
- Tarcai, N., Virágh, C., Ábel, D., Nagy, M., Várkonyi, P. L., Vásárhelyi, G., and Vicsek, T. (2011). Patterns, transitions and the role of leaders in the collective dynamics of a simple robotic flock. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(04):P04010.

Numerical Artificial Chemistries

Juan Camilo Ramírez, James Marshall

Kroto Research Institute, Department of Computer Science, University of Sheffield, UK

J.Ramirez@dcs.shef.ac.uk, James.Marshall@sheffield.ac.uk

Extended Abstract

A classic problem in evolutionary biology is to investigate the emergence of life-related phenomena from the interaction of non-living agents and explore how those aggregated interactions give way to the emergence of pre-biotic structures. Artificial chemistries offer a way to model rudimentary ecosystems where elementary properties or processes of life can emerge only as the result of the interaction of simpler components with no individual complex behaviour. These emergent phenomena can range from the cooperative co-evolution of mutually-catalysing compounds (Fishkis, 2011) to self-replicating cell-like structures (Hutton, 2007). The ease to obtain such results is directly linked to the nature and representation of the artificial chemistry used because this limits the type of novelty that can emerge from these rules.

We classify chemistries by the type of reactions they allow. In certain chemistries, chemical species are transformed into different species by arbitrary rules; this allows dynamical systems analysis of convergence or divergence but does not explicitly consider composition of atoms into molecules. Organisation theory (Dittrich, 2007) is mostly based on this type of chemistries. On the other hand, physical novelty can be achieved by allowing new structures to emerge by allowing sets of atoms to bond together or break apart. Results with this type of chemistry include Hutton's (Hutton, 2007).

We propose a representation of an artificial chemistry as follows. A set of molecules represented as integers with prime numbers acting as their indivisible constituent atoms and a set of reactions that occur only when the potential reactants are near each other and their numeric values satisfy conditions defined as mathematical functions. Products are calculated as functions of the reactants and consist of rearrangements of their prime factors. In this manner, reactions result in the transfer of atoms between molecules, similarly to reactions in nature. The set of molecules is therefore no longer necessarily finite and reactions can be defined to apply to a family of molecules rather than to specific molecules themselves. This approach is proposed in order to have the advantages of other artificial chemistries and explore emerging patterns with greater generality.

We use this chemistry to model a reaction soup whose composition evolves with repeated application of a family of reactions and determine the conditions where this iterative process makes species diversity drift indefinitely, or converge to a stable point where all species remain with the same concentrations. We show how this latter case occurs when the remaining species are coexisting by cyclically promoting the replication of each other.

References

- Fishkis, M. (2011). Emergence of Self-Reproduction in Cooperative Chemical Evolution of Prebiological Molecules. *Origins of Life and Evolution of Biospheres*, 41:261-275.
- Hutton, T. J. (2007). Evolvable self-reproducing cells in a two-dimensional artificial chemistry. *Artificial Life*, 13(1):11–30. MIT Press.
- Dittrich, P., & di Fenizio, P. S. (2007). Chemical organisation theory. *Bulletin of mathematical biology*, 69(4):1199–1231. Springer.

Dynamic phase transition in a system of self-propelled particles

Maksym Romensky, Vladimir Lobaskin

School of Physics and Complex and Adaptive Systems Lab, University College Dublin, Belfield, Dublin 4, Ireland
vladimir.lobaskin@ucd.ie

Extended Abstract

Collective motion phenomena in nature can be observed at extremely diverse scales, ranging from single cells and unicellular organisms (Ben-Jacob (1994)) up to higher organisms (Rauch et al. (1995); Weihs (1973)), such as insects, birds, fish, or even mammals. It is now clear (Vicsek et al. (1995); Chaté et al. (2008); Couzin et al. (2002, 2005)) that collective motion in systems of active agents arises from simple rules followed by individuals and involves neither central coordination nor external field. Self-propelled particles models (SPP) are now commonly used for studying of the collective dynamics in biological systems. One of the simplest possible SPP models exhibiting collective behaviour was proposed by Vicsek and coworkers (Vicsek et al. (1995)). Since then the original model has undergone further development and modification (Chaté et al. (2008); Couzin et al. (2002, 2005)), which allowed one to reproduce multiple types of collective behaviour. It has been found that the transition to self-organized behaviour occurs at sufficiently high density of active particles and sufficiently high ratio of driving power to dissipation power (Chaté et al. (2008)) and requires an attractive or an aligning interaction between the particles. Although the onset of the collective motion has been well studied, the stability of the collective mode with respect to variation of the main characteristics of motion and interaction is not known.

We use a Vicsek-type model (Vicsek et al. (1995)) to study effects of particle finite size and non-thermal (behavioural) noise on transport and ordering in the unbounded system of self-propelled particles. Our two-dimensional model consists of point particles moving at constant speed inside a square simulation box with periodic boundary conditions. The directions of motion of individual particles, determined simultaneously during each time step, are affected by interactions with other particles located within one of two non-overlapping circular neighbourhoods centred on each particle. The first interaction zone or radius $r_1=1$, which can also be considered as a particle size, is responsible for the repulsion. If other agents are present within this zone the particle moves away from the center of mass of the particles in the circle. If no agents are found within the first circle particle responds to the next zone - zone of alignment, which in our study had size $r_a=5$. The particle assumes the average direction of motion of other particles in this neighbourhood with some added uncertainty, which is specified by a random turning angle drawn from a Gaussian distribution. We analysed the motion statistics for particles in the steady state regime using velocity autocorrelation and velocity spatial correlation functions. We measured the orientational order parameter, ϕ , where ϕ is the ensemble average of the particle velocity vector, as a function of intensity of internal noise and particle density. This order parameter turns zero in the disordered phase and takes non-zero values up to $\phi \approx 0.92$ in the ordered phase.

Our results show that in addition to the transition to a dynamically ordered state on increasing the particle density, as reported previously (Chaté (2008); Couzin (2002); Gönci (2008)), there exists a re-entrant transition into a disordered phase at the higher densities. The re-entrant disordered behaviour at high densities can be associated with the repulsions between the particles, which destroys the ordering in the crowded state upon increasing frequency of collisions. The repulsions play an important role also in the absence of noise, as they prevent formation of a perfectly ordered phase. Moreover, one can also see that the presence of non-thermal noise significantly narrows the region of the ordered behaviour. The transition into an orientationally ordered state is possible only after reaching certain levels of the noise strength and density. The dynamical phase diagram reflecting the behavior of the order parameter for various values of ρ and η is shown in fig. 1. The plot confirms the existence of the optimum intervals for both density and the magnitude of noise within which the global ordering (non-zero average ϕ) is possible. The minimum density required for the ordering is quite small. In dense systems, the range of the ordered behavior is widest at $\eta = 0$ and is getting narrower upon increasing η . We also note that the density, at which the maximum ordering is observed, is increasing from $\rho \approx 0.05$ at small $\eta=0$ to $\rho \approx 0.23$ at $\eta=0.92$. The stationary value of the order parameter in each configuration is determined by the competition between the disordering noise and the aligning interactions. Although the noise in our model is non-thermal, it plays the same role as temperature in the order-disorder transitions (similar to ferromagnetic phase transition). One can see that at the values of noise $\eta \approx 0.92$, which correspond to the turning angle of 60° , the ordering is no more possible.

It is interesting to see how the ordering changes close to and at the transition point. In our system, the orientational order parameter varies continuously across the transition on varying the particle density at the fixed noise level. Similarly, the transformation happening on increasing the noise level, when the density is fixed, is continuous with the order parameter decaying to zero. Another important observation, consistent with the phenomenology of phase transitions in condensed

matter systems (Landau (1980)), is the change of the character of order parameter fluctuations. We find that in the disordered phase, the orientational correlations of particle velocities decay exponentially, so that the size of the correlated domain is finite. The correlation radius is increasing upon approaching the transition point. At the transition point, as well as in the ordered phase we observe long-range power law decay of the correlations, which enables a formation of large scale ordered swarms.

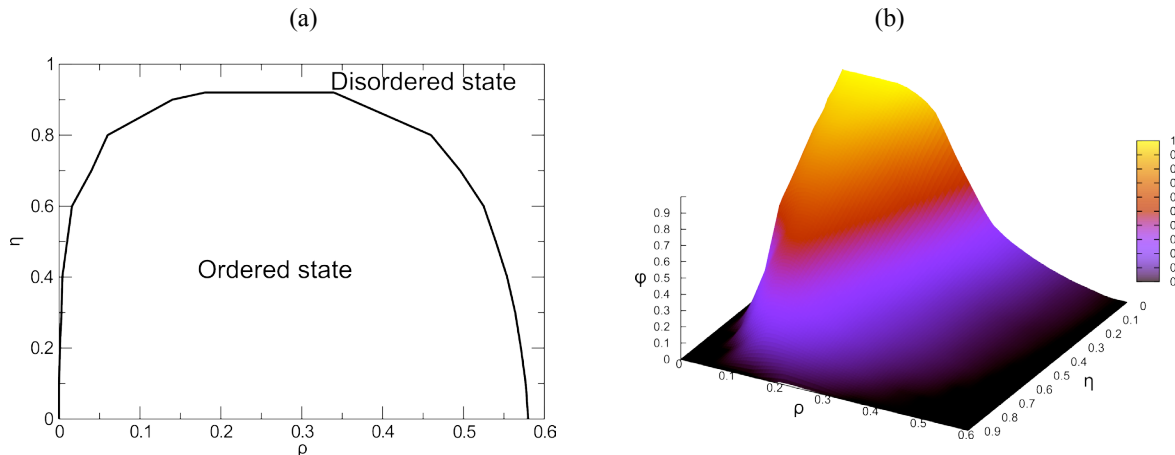


Figure 1: Dynamic self-organization map in the Vicsek-type model upon variation of particle number density ρ and noise level η . (a) The ordered state corresponds to a non-zero average order parameter or a net drift of the swarm. Disordered state refers to an absence of the net drift. (b) The height of the surface ϕ reflects the amount of orientational order in the system: corresponds to perfectly aligned velocities across the system, 0 to a disordered system.

Acknowledgments

This research was supported by the Irish Research Council for Science, Engineering and Technology.

References

- Ben-Jacob, E., Schochet, O., Tenenbaum, A., Cohen, I., Czirók, A., and Vicsek, T. (1994). Generic modelling of cooperative growth patterns in bacterial colonies. *Nature*, 368:46-49.
- Chaté, H., Ginelli, F., Grégoire, G., Peruani, F., and Raynaud, F. (2008). Modeling collective motion: variations on the Vicsek model. *Eur. Phys. J. B - Condensed Matter*, 64(3/4): 451-456.
- Chaté, H., Ginelli, F., Grégoire, G., and Raynaud, F. (2008). Collective motion of self-propelled particles interacting without cohesion. *Phys. Rev. E*, 77: 046113.
- Couzin, I. D., Krause, J., Franks, N. R., and Levin, S. A. (2005). Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025): 513-516.
- Couzin, I. D., Krause, J., James, R., Ruxton, G. D., and Franks, N. R. (2002). Collective memory and spatial sorting in animal groups. *J. Theor. Biol.*, 218:1-11.
- Gönci, B., Nagy, M., and Vicsek, T. (2008). Phase transition in the scalar noise model of collective motion in three dimensions. *Eur. Phys. J. Special Topics*, 157:53-59.
- Landau, L. D. and Lifshitz, E.M. (1980). *Course of Theoretical Physics*, vol. 5 (Butterworth-Heinemann, 3rd ed.)
- Rauch, E. M., Millonas, M. M., Chialvo D. R. (1995). Pattern formation and functionality in swarm models. *Phys. Lett. A*, 207:185-193.
- Vicsek, T., Czirok, A., Ben-Jacob, E., Cohen, I., and Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.*, 75:1226-1229.
- Weihs, D. (1973). Hydromechanics of Fish Schooling. *Nature*, 241:290-291.

Evolutionary Swarm Chemistry in Three Dimensions

Hiroki Sayama

Binghamton University, Binghamton, NY 13902-6000
sayama@binghamton.edu

Extended Abstract

We extended the Evolutionary Swarm Chemistry model (Sayama, 2011a,b) to three-dimensional (3D) space and compared its behavior with the original two-dimensional (2D) version. The purpose of this study was to investigate the influence of spatial dimensionality on evolutionary dynamics of swarms.

Swarm Chemistry (Sayama, 2009) is an artificial chemistry framework that can demonstrate self-organization of dynamic patterns of kinetically interacting heterogeneous particles. A swarm population in Swarm Chemistry consists of a number of simple self-propelled particles moving in a continuous 2D space. Each particle can perceive average positions and velocities of other particles within its local perception range, and change its velocity in discrete time steps according to kinetic rules similar to those of Reynolds' Boids (Reynolds, 1987). Each particle is assigned with its own kinetic parameter settings (similar to genotype) that specify preferred speed, local perception range, and strength of each kinetic rule. Particles that share the same set of kinetic parameter settings are considered of the same type.

Several model extensions have been introduced to the model, including local information transmission among particles and their stochastic differentiation/re-differentiation. These extensions made the model capable of showing morphogenesis and self-repair (Sayama, 2010) and autonomous ecological/evolutionary behaviors of self-organized "super-organisms" made of a number of swarming particles (Sayama, 2011a). We also have recently extended the original non-evolutionary Swarm Chemistry model into a 3D space and studied the robustness of swarm morphologies against dimensional changes (Sayama, 2012).

Here, we applied the same dimensional upgrade to the evolutionary version of Swarm Chemistry. Aside from spatial dimensions, almost all other model assumptions and parameter settings were carried over as is from the 2D version. The only parameter changed was the length of the side of space. In the original 2D model, the space was a continuous square whose side length was 5000 (in arbitrary units). In 3D, the space is a continuous cube with side length 2000. With the same number of particles (10000), this set-

ting makes the average number of neighbor particles within an interaction range of a typical particle about the same in both 2D and 3D. We also tested spatial sizes larger and smaller than this, but the swarm behavior in 3D appeared most similar to 2D when the side length was 2000.

We conducted simulations of swarm evolution using the parameter settings above. The initial configuration was 100 active particles with randomly generated genomes, together with 9900 passive particles, uniformly and randomly distributed over the cubic 3D space. We used the "revised-high" experimental condition with high mutation rates and dynamic environmental changes, which was previously identified as most successful in maintaining continuous evolutionary exploration without losing macroscopic structures (Sayama, 2011b).

A typical simulation run is shown in Fig. 1, in comparison with 2D results. While the behaviors of swarms in 3D were certainly interesting, it was immediately realized, to our surprise, that their dynamics were not as creative and evolutionary as their 2D counterparts. We tested several parameter variations but this general observation remained the same.

To confirm this intuitive observation, the evolutionary exploration activity of swarms was measured using the method introduced in (Sayama, 2011b). Figure 2 shows the results for both 2D and 3D, presenting a clear difference between the two spatial settings. Most notable is that the repeated sharp peaks, i.e., rapid productions of new particle types due to environmental changes, are missing in the 3D cases. Since environmental perturbations are implemented as a temporary change of selection criteria applied to competing kinetic rules on colliding particles, the lack of peaks means that collisions of particles are not occurring in 3D as frequently as in 2D, causing low selection pressure and slow evolutionary progresses.

We interpret that this qualitative difference was partly due to the well-known fact that the probability for a random walk particle to come back to its origin drops below one if the spatial dimensions are increased from two to three (Pólya, 1921; Domb, 1954), which also makes the likelihood of par-

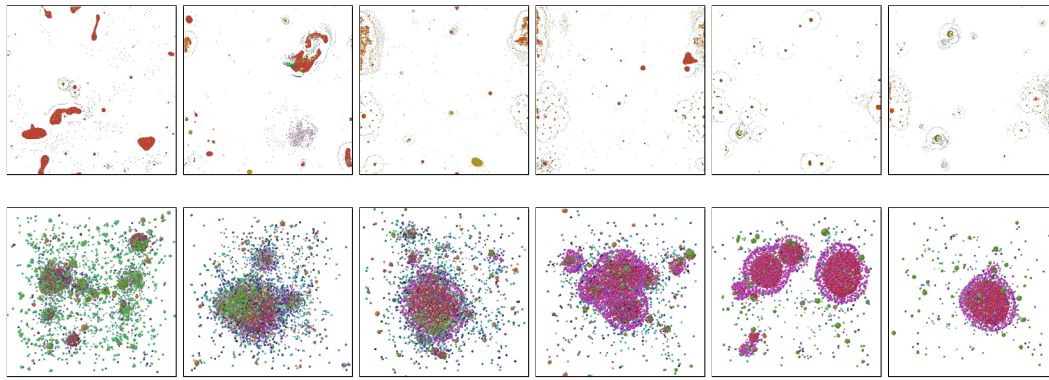


Figure 1: Typical simulation runs. Time flows from left to right. Top: 2D version. Bottom: 3D version. It may be hard to notice differences in the swarm's dynamical behavior on these static pictures. We encourage readers to watch movies of simulation results online at <http://youtube.com/complexsystem>.

ticle collisions much smaller and thereby slows down the morphogenesis and evolution of swarms in 3D.

Our finding that evolution of swarms is highly sensitive to dimensional changes marks a stark contrast to our other finding that their self-organization is highly robust against the same changes (Sayama, 2012). Moreover, our results may also indicate a general principle that biological evolution takes place much more efficiently on a 2D surface than in a 3D space, because the former environment allows organisms to encounter each other more frequently and thereby facilitates their competition and selection. This could lead us to some interesting speculations, e.g., that evolution in 3D would only flourish after organisms acquired abilities of long-range perception and active chasing to ensure their frequent encounters, and also that there may not be any large-scale lifeform evolving in a vast 3D interstellar space.

References

- Sayama, H. (2011a). Seeking open-ended evolution in Swarm Chemistry. *Proc. IEEE ALIFE 2011*, IEEE, pp.186-193.
- Sayama H. and Wong, C. (2011b). Quantifying evolutionary dynamics of Swarm Chemistry. *Proc. ECAL 2011*, MIT Press, pp.729-730.
- Sayama, H. (2009). Swarm chemistry. *Artif. Life* 15:105-114.
- Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics* 21(4):25-34.
- Sayama, H. (2010). Robust morphogenesis of robotic swarms. *IEEE Comp. Intell. Mag.* 5(3):43-49.
- Sayama, H. (2012). Morphogenesis of self-organizing swarms in 3D Swarm Chemistry. *Proc. GECCO 2012*, in press.
- Pólya, G. (1921). Über eine Aufgabe der Wahrscheinlichkeitsrechnung betreffend die Irrfahrt im Straßennetz. *Mathematische Annalen* 84:149-160.
- Domb, C. (1954). On multiple returns in the random-walk problem. *Math. Proc. Camb. Phil. Soc.* 50:586-591.

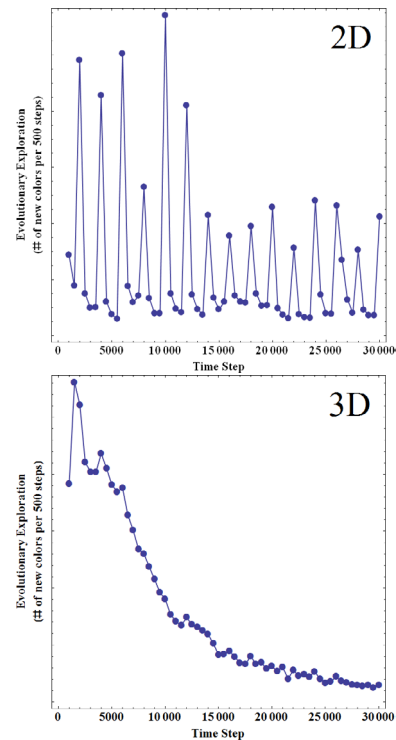


Figure 2: Temporal changes of the evolutionary exploration measurement, i.e., number of new colors that appeared in every 500 time steps (Sayama, 2011b). Top: 2D, Bottom: 3D. Each curve shows average results over three independent runs starting with random initial conditions. The scale of vertical axes depends on the visualization method, which was different between 2D and 3D models, and therefore the two plots are not comparable quantitatively. Instead, we compare chronological patterns of evolutionary exploration. Sharp spikes seen in 2D (and at the beginning of 3D) were due to dynamic environmental changes.

Behavior and Intelligence

Extended Abstracts

Sustainable Population of Autonomous Foragers in a 3D environment with Physics

Nicolas Chaumont^{1,2}, Christoph Adami^{2,3}

¹Keck Graduate Institute of Applied Life Sciences, Claremont, CA 91711

²BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, MI 48824

³Microbiology & Molecular Genetics, Michigan State University, East Lansing, MI 48824
adami@msu.edu

Extended Abstract

In order to observe the dynamics of open-ended evolution and adaptation of virtual agents simulated within a 3D environment, the acquisition of energy (via foraging) is the single most important necessary capacity. Evolving this foraging behavior within an open-ended environment, however, is a difficult task because a starving population will go extinct before any agent has developed the capacity to forage. To circumvent this problem, we have evolved foraging behavior outside of the population using a standard Genetic Algorithm (GA), and placed the capable organisms into a world in which energy is limited. We observed that the resulting population is stable and can sustain itself by foraging and consuming the limited resource. Such populations can be used to study the evolution of more sophisticated foraging strategies, to adapt these strategies to resources of different types (adaptive radiation) and to respond to geographical as well as morphological variance, without the need of an external (and arbitrary) fitness function.

We understand open-ended environments as physical spaces in which organisms can live, and potentially replicate if they gathered a sufficient amount of energy. Evolution in open-ended environments is different from evolution in a GA as replication is not automatic, nor is there an explicit fitness function that assigns fitness to a genotype. In open-ended environments, fitness is implicit and can only be assessed in hindsight for those types that have been able to persist for long periods, just as for biological organisms. In such simulations, individuals accumulate energy by reaching food items, and produce a clone if their energy reaches a reproduction threshold. The forager consumes energy at a rate that is proportional to its volume (to maintain its metabolic function), and is removed if its energy level drops below a starvation threshold. The virtual organisms used in this work are inspired by Sims (1994) and similar to the blocky walkers used in Chaumont et al. (2007), but have two additional sensors: one that returns the angle and another the distance to the closest food source. In this abstract, we briefly describe the strategy we used to evolve foragers capable to sustain a population in an open-ended environment, and then present the first results for an ecol-

ogy of foragers. The present system adds two features that do not exist in standard ecological simulations: first, the organism's controller and morphology are co-evolved *de novo* and second, the organisms are subject to a realistic physical environment, creating a rich adaptive landscape. Such simulations can complement other tools used in studies where the animal's motion capacity plays an important role.

The foragers used in this work were evolved with a steady-state Genetic Algorithms (SSGA), in a multi-stage method where each stage provides conditions favorable to the emergence of intermediate, increasingly elaborate skills that build upon each other to ultimately yield dependable continuous foraging. A stage consists of many replicates of SSGAs (similar to those used in Chaumont et al. 2007), identical except for the random seed. Each replicate (a population of 200 organisms evolved for a fixed number of generations) yields an evolved organism that is inspected to assess its performance against a selection criterion that is stage-dependent (Table 1). Only one individual from the current stage, called a *key organism*, is used to seed all the replicates in the next stage: this seeding is called a "transfer". The larger number of replicates in the first two stages was necessary to obtain at least five suitable candidates for transfer. New stages lead to improvements either through perfecting existing skills, or through the emergence of new ones.

The design of a fitness landscape that leads to the evolution of a desired character is often more art than science. Here, we have converged on a number of conditions necessary for the emergence of key behavioral milestones through a process of trial and error. These conditions fall into two categories: 1) GA parameters: number of generations, fitness function, selection regime, 2) Initial environmental conditions: food source position pattern, noise level. The fitness function used in the first three stages (see Table 1) favors locomotion (W_l), food source approach (W_s), and reaching targets (W_r), and is designed for roulette selection for the first two food sources. A detailed explanation of each term and their rationale is provided in Chaumont and Adami (2011). After organisms can reach two targets in sequence, the fitness in a population varies so much that

stage	replicates	generations	noise level	fitness function	selection criterion for the key organism
1	400	40	0.1%	base*	reliable steering towards at least 3 directions
2	480	50	5%	base*	reliable approach of at least 3 food sources
3	100	50	50%	base*	reach all the food sources placed on an 11x11 grid
4	120	50	100%	reach > 2 targets	reach 6 food sources in sequence the most often
5	138	50	100%	# targets reached	reach 10 food sources in sequence the most often
6	144	50	100%	# targets reached	reach 10 food sources in sequence the most often

Table 1: **Parameters used at each stage.** A stage embodies a set of environmental conditions that favors the emergence of a given skill that takes the organism closer to dependable foraging. * See Chaumont and Adami (2011) for a detailed description.

diversity can be lost when roulette selection is used. Instead, starting with stage 4 we use tournament selection, while omitting the reward for reaching the first food source. Food sources are located in the four cardinal directions and placed 10 meters away to encourage steering, and organisms are scored on their ability to move in each of the four directions. For each direction, there is a chance to obtain two more food sources placed in the same direction that appear at the same distance, also sequentially. To prevent over-adaptation to the direction, we add noise to each target position that we increase in later stages as the foragers become more capable (Table 1). Evolving foragers with greater amounts of noise is too challenging, and gives rise to strategies that fail to react to the target positions.

After 290 generations (across 1382 runs), the final foragers were able to reach the first and subsequent food sources about 95% of the time. To test whether our evolved foragers could form stable populations, we carried out “ecological” simulations that were seeded with an initial population of 16 identical organisms positioned uniformly randomly on a square surface of 160x160 meters. Note that this environment is very different from the one the foragers were exposed to in the GA, as in the open-ended environment foragers face for the first time other foragers that may reach and absorb food that they themselves had targeted. With a constant influx of energy and seeded with capable foragers, the environment reaches its carrying capacity and individuals compete for a limited amount of resources¹. In the environment depicted in Figure 1 (top), the food sources decay slowly (but exponentially) and disappear if they are absorbed. To determine whether a population is stable, it is simulated for an amount of time that is two orders of magnitude longer than an organism’s lifespan when starved. (Fig.1 bottom). After an initial transition period when the population is small, food accumulates in the world and triggers an exponential population growth. Eventually, the population stabilizes and maintains healthy levels throughout the simulation, as in a standard chemostat. This simulation environment provides a basic platform to study the evolution of virtual organisms in their 3D physical environment.

¹A video is available at <http://youtu.be/3eTzciBz2VY>

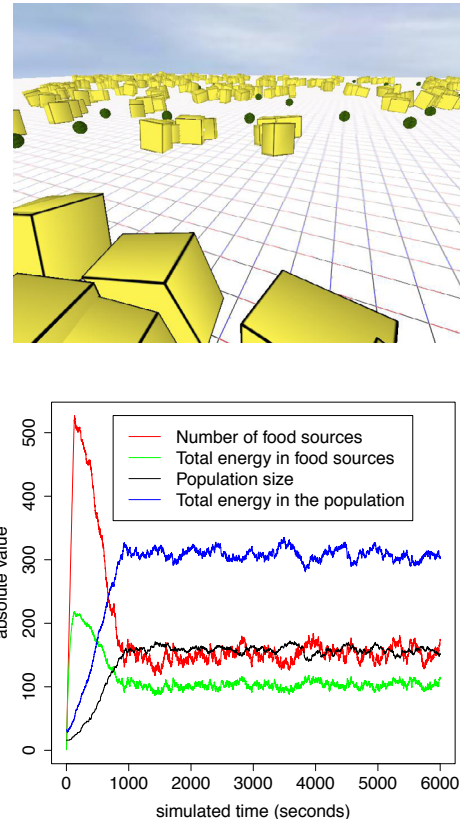


Figure 1: **A simulated ecology:** A snapshot of a population of foragers (yellow) foraging for green food sources (top). A graph showing the food source-dependent population size as well as resource abundance and internal energy (bottom).

References

- Chaumont, N. and Adami, C. (2011). Evolution of sustained foraging in 3D environments with physics. <http://arxiv.org/abs/1112.5116>.
- Chaumont, N., Egli, R., and Adami, C. (2007). Evolving virtual creatures and catapults. *Artificial Life*, 13:139–157.
- Sims, K. (1994). Evolving virtual creatures. In *SIGGRAPH '94*, pages 15–22, New York, NY, USA. ACM.

Mapping the collective intelligence of the Artificial Life XIII stakeholders

Mark Dorr¹, Sif Schmidt-Petersen¹, Harold Fellermann¹, Lone Laursen¹ and Steen Rasmussen^{1,2}

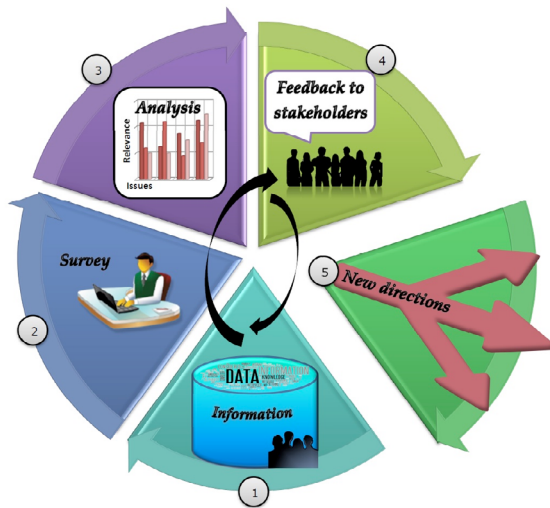
¹Center for Fundamental Living Technology (FLinT), University of Southern Denmark, 5230 Odense, Denmark

²Santa Fe Institute, 1399 Hyde Park Rd, Santa Fe NM 87501, USA
steen@sdu.dk

Extended Abstract

The information and communication technology (ICT) based collective intelligence method we present stem from several different traditions including the facilitation of citizen participation, the study of social groups, the use and development of survey methods, automated language parsing as well as investigation of the many new web 2.0 possibilities. Our first web-based method was used to gauge the collective intelligence of the Artificial Life community in connection to the Artificial Life VII Conference in Portland, summer 2000 (Rasmussen et al., 2003). Here we propose to utilize our newly developed method to again gauge the collective intelligence of the Artificial Life Community, but now 13 years later in connection to the upcoming Artificial Life XIII Conference. We propose to set up a survey experiment during the Artificial Life XIII Conference, where we ask the community what they think about their own challenges, successes and failures. Due to our automated feedback system combined with language parsing, we expect to be able to provide part of the results to the community already by the end of the Artificial Life XIII Conference.

The method works through a series of steps, as illustrated in the figure below. The method is able to map the “lay of the land” for any complex set of issues within a large stakeholder community. The method is fast and inexpensive as all input comes from an online survey interface where the stakeholder community defines their own issues.



1. A small, diverse, and representative subset of the stakeholder group designs an initial information repository, including formulating key questions about the problem complex on the Web. In the current situation, this small subset of the stakeholder group consists of the authors of this text together with our artificial life research team at FLinT. Next all stakeholders in the larger community individually review the information about the issue complex, either through the associated Web environment or through town hall meetings, media, conversations, and so on. Where possible, stakeholders add information about the problem context to the Web storehouse for review by others.

2. All stakeholders in the group rank and organize issues relevant to the problem and express their opinions about the issues through an online, open-response survey that allows freely typed input to questions. Individuals can describe new issues as well as rank possible already defined issues.

3. The feedback from step 2 is parsed, synthesized and analyzed to identify possible areas of conflict and consensus via graphical frequencies, a variety of statistical correlations, mind maps, and other relevant plots. This analysis can increasingly be done automatically

online by using an off the shelf language parser to extract the pertinent concepts from the open text input provided by the stakeholders. These concepts can then automatically go through statistical analysis and eventually be graphed.

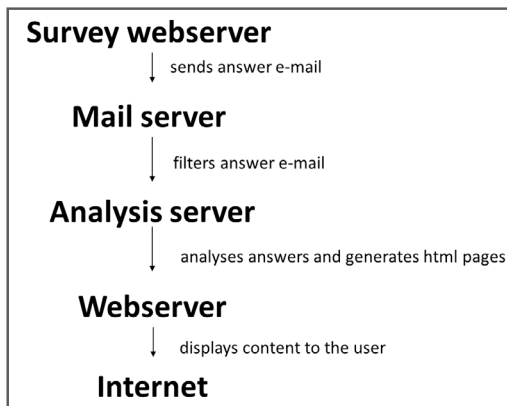
4. The results of the analysis, the graphs and statistical findings, are made available to the stakeholders at large through the Web.

5. The condensed collective intelligence now gathered in the data repository, through the above-described process, allows the whole community to make decisions based on the information resulting from the analysis of all the answers from the survey. The process is transparent and bottom up.

Steps 2–4 can be repeated as the group reacts to areas of conflict and agreement, and as individuals modify their positions. Once a group has clarified its conflicts and identified its areas of consensus, it can take action on these matters. It is our contention that this sort of self-organizing collective intelligence process enables a group to make better-informed decisions about the important

problem complexes that it faces. Some of the questions we propose to address by asking the Artificial Life community include two to four accomplishments, weaknesses, scientific challenges, related scientific communities as well as engineering applications. Also, input regarding envisioned potential societal impact positive as well as negative will be requested. Together with demographic information (age, sex, field, position, country, etc) we should be able to provide a more detailed understanding of the current lay of the land within our community.

A special aspect of the collective-intelligence process described above is the open-ended responses allowed in step 2. The open-response survey does three important things: It organizes stakeholder input along a set of broadly defined questions about the set of issues; it allows open-ended input; and it limits each response of each individual to a few sentences. Once the open-response data has been gathered, it is brought into a quantifiable form by coding each response into one of a finite number of response categories either done manually or via statistical methods. We use the python programming language together with the open source library called the Natural Language Toolkit (NLTK, <http://www.nltk.org>), see Bird, Klein and Loper, 2009, to parse the text and identify the key concepts or concept combinations. The NLTK is an easy-to-use, well documented concept, and its in-line script processing is well in sync with the workflow of the server infrastructure system. When doing computational linguistics, or natural language parsing, there are a number of ways to process a body of text. The NLTK allows us to tokenize a text, i.e. splitting it into correct grammatical categories; stemnize, that is, reducing verbs to their stem-form; filter the text and ask for specific categories of words; and by use of the semantic module it also makes it possible to extract meaning out of a text.



The data analysis flow and server infrastructure, as shown in the figure to the left, consist of five parts: (i) survey webserver for question presentation, with user management (tracks user activities), (ii) mail server for message forwarding, sorting and backup, (iii) analysis server for running analysis software in a secured environment (bash, python (jinja2, matplotlib, nltk), c++, java,...), (iv) webserver for displaying the html content, (v) internet for distributing the content.

Familiar statistical and other methods (e.g. support vector machines and Bayesian filters) can be employed to extract information from the open-response data once it has been categorized.

The key feature of the open-response survey is the ability to take input that is completely open in content and restricted only in length. The open-response survey can be thought of as a “fishing net” that efficiently and inexpensively catches all the worries, excitement, visions, complaints, and the like in the

group and makes them available for both qualitative and quantitative analysis. This mitigates the familiar bias in traditional surveys caused by forcing all responses to be chosen from predefined answers to predefined questions.

We intent to publish the results of the proposed Artificial Life XIII survey in the Artificial Life journal.

References

- S. Rasmussen, M. Raven, G. Keating and M. Bedau, Collective intelligence of the Artificial Life community on its own successes, failures and future. *Artificial Life* 9: 207–235 (2003).
- S. Bird, E. Klein and E. Loper, Natural Language Processing with Python, Analyzing Text with the Natural Language Toolkit, O'Reilly Media, June 2009
- <http://www.nltk.org/>

An Analysis of the De Novo Evolution of a Complex Odometric Behavior

Laura M. Grabowski¹, David M. Bryson², Fred C. Dyer², Robert T. Pennock², and Charles Ofria²

¹University of Texas-Pan American, Edinburg, TX 78539

²BEACON Center for the Study of Evolution in Action
Michigan State University, East Lansing, MI 48842
grabowskilm@utpa.edu

Extended Abstract

Many researchers are interested in the evolution of biological complexity. Animal behavior provides many examples of complexity, but investigating how animal behavior evolves is difficult. The fossil record leaves few clues that would allow us to recapitulate the path that evolution took to build a complex behavior, and the long time scales required prevent us from re-evolving such behaviors in a laboratory setting. Studying the evolution of behavior in a virtual world offers the opportunity to observe evolving behavior, from simple capabilities to complex behavioral repertoires. We can also analyze the relationship of the underlying genetic structure to behavior, an option that is available only rarely with living creatures.

We present results of a study in which digital organisms—self-replicating computer programs that are subject to mutations and selection—evolved in different environments that required information about past experience for fitness-enhancing decisions. One population evolved a mechanism for step-counting, a surprisingly complex odometric behavior that was only indirectly related to enhancing fitness. This behavior arose in open-ended evolution in experiments conducted in the Avida system (Ofria et al., 2009), in an environment where neither counting nor distance tracking were directly selected for.

Our experimental environments were inspired by maze-learning experiments with honey bees (Zhang et al., 1996), where the bees learned to follow different visual cues through a maze to a food goal. We designed our experiments to explore the evolution of memory use. Effective strategies in these environments involved different ways of storing and reusing experience from the individual's lifetime (Grabowski et al., 2010, 2011). During evolution, organisms were presented with a randomly selected path formed by sensory cues in the environment. In the experiments that produced our case study organism, each path contained only right turns or left turns. Our case study organism exhibited an unusual backtracking behavior when traversing right-turn paths. Earlier analysis (Grabowski et al., 2011) revealed that the organism counts the number of steps it has moved on a right-turn path and turns around at a specific point. The current discussion traces how this complex behavior arose during evolution.

Site	Instruction	Instruction Functionality
117	h-search	Marks the start of counting module.
118	sg-rotate-r	Turn right 45°
119	if-grt-0	CX register contents > 0?
120	nop-C	This comparison is TRUE when on a right-turn path.
121	h-copy	Copy (executes only when on a right-turn path).
122	h-copy	Copy (always executes).
123	sg-sense	Put current sense input into CX register.
124	nop-C	
125	jmp-head	Move the IP the number of instructions designated by the value in CX register. If sense input was nutrient, CX = 0; IP does not change. If sense input was right, CX = 2; IP skips 126-127 and moves to site 128. If sense input was left, CX = 4; IP skips 126-129 and moves to site 130.
126	sg-rotate-l	Executes only when sense input is nutrient (<i>i.e.</i> , CX = 0). When executed, undoes right turn at top of loop (site 118).
127	if-equ-X	BX register contents = 1? This comparison is TRUE on a right-turn path.
128	get-head	Put the current value of IP into CX (<i>i.e.</i> , CX = 128). Executes only when on a right-turn path.
129	sg-move	Take a step. Does not execute on a left-turn path.
130	inc	Increment value in BX (<i>i.e.</i> , BX = BX + 1).
131	if-n-equ	If contents of BX are not equal to contents of CX, execute the next instruction. Instruction tests for loop exit conditions.
132	mov-head	Exit on right-turn path after taking 127 steps, then incrementing BX to 128. Exit on left-turn path after turning 180° (4 1/8 turns) without taking any steps.

Table 1: Detail of evolved step-counting organism's genome, listing instructions in the counting module and the effects of executing the instructions in right- and left-turn environments. Instructions highlighted in green mark the beginning and end of the enclosing loop (sites 117 and 132); instructions highlighted in yellow (sites 123-125) determine which of the subsequent instructions will execute, based on the currently sensory information; instructions highlighted in blue perform the counting and control loop exit.

The evolved step-counting mechanism provides an excellent example of how complexity evolves. The step-counter was built from the inside out, by assembling two separate instruction sequences and the loop that ultimately contained them. One of the sequences functions to selectively execute instructions according to current environmental conditions, and the other sequence

contains the counter controls iterations of the enclosing loop. Table 1 shows a detailed listing of the step-counting instructions in the organism's genome. These components were later critical to the operation of the complex step-counting feature, but sometimes arose without conferring any immediate fitness benefit, or were even initially deleterious. At some periods of evolution, there were sudden dramatic improvements in fitness or performance, compared to other periods where evolution slowly fine-tuned a trait. These results are consistent with theoretical views about how complexity evolves, and demonstrate how complex behavioral traits can arise even in very simple environments without direct selection.

References

- Grabowski, L. M., Bryson, D. M., Dyer, F. C., Ofria, C., and Pennock, R. T. (2010). Early evolution of memory usage in digital organisms. In *Artificial Life XII: Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems*, pages 224–31, Cambridge, MA. MIT Press.
- Grabowski, L. M., Bryson, D. M., Dyer, F. C., Pennock, R. T., and Ofria, C. (2011). Clever creatures: case studies of evolved digital organisms. In *Proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems (ECAL 2011)*, pages 276–283, Cambridge, MA. MIT Press.
- Ofria, C., Bryson, D. M., and Wilke, C. O. (2009). Artificial life models in software. In Adamatzky, A. and Komosinski, M., editors, *Advances in Artificial Life*, chapter Avida: A Software Platform for Research in Computational Evolutionary Biology, pages 3–36. Springer-Verlag, Berlin, 2nd edition.
- Zhang, S. W., Bartsch, K., and Srinivasan, M. V. (1996). Maze learning by honeybees. *Neurobiology of Learning and Memory*, 66:267–282.

The Role of Local and Global Perspectives in the Dynamics of Opinion Convergence and Polarization

Patrick Grim,¹ Aaron Bramson,^{2,3} Daniel J. Singer,³ Steven Fisher,² Carissa Flocken⁵ and William Berger^{2,6}

¹Group for Logic & Formal Semantics, Philosophy, Stony Brook University

²Center for Study of Complex Systems, University of Michigan

³Department of Philosophy, University of Michigan

⁵Honors Program, University of Michigan

⁶ Department of Political Science, University of Michigan

pgrim@notes.cc.sunysb.edu

Extended Abstract

What are the crucial factors governing the phenomena of belief polarization and convergence in a population? Popular opinion has it that polarization in American has increased dramatically (Brownstein 2007, CBS poll). Much of the academic literature takes a more nuanced view (Fiorina 2010), though even there we see the need for more careful attention to different senses of the term. What exactly does belief polarization consist in, and how can we measure it? Our target is a better understanding of the range of phenomena that fall under the term, both (a) conceptually and (b) through study of polarization dynamics in an agent-based network model of belief updating.

We model beliefs as real values between 0 and 1, using agents embedded in random communication networks. Figure 1 shows an initial configuration.

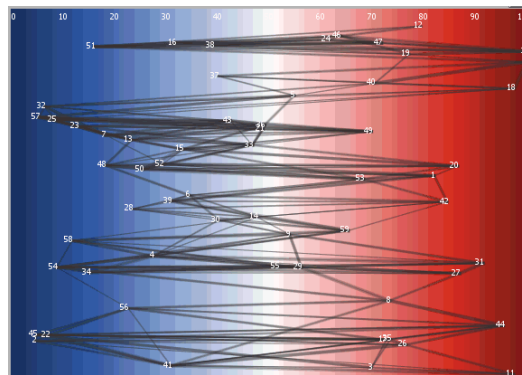


Fig. 1 Initial randomization of beliefs and network connections

At each stage of the simulation, agents update both their belief and their level of trust in those with whom they are linked. Belief is updated on the model of reinforcement: beliefs are more strongly reinforced by information from contacts one trusts (Visser & Cooper 2003). In our model, trust levels are represented by weighted links, and agents' beliefs are updated using a weighted averaging of the beliefs of network contacts.

But it is also true that widely divergent opinions can strain bonds of trust (Lord, Ross & Lepper 1979). We use a linear trust update that increases or decreases an agent's trust in neighbors based on their difference in belief. The hypothesis is that we can more fully understand the dynamics of belief polarization in terms of the interplay between (a) belief revised in terms of trust and (b) trust revised in terms of belief.

A main target in this first study is the role of global and local perspectives in trust updating. In global updating, our linear trust function is applied to the range of beliefs across the entire population. Trust is increased linearly for those within a distance of τ from an agent's belief (.1, .2, .3...). It is decreased linearly for those beyond that distance. In local updating, that scale is determined only by the range of beliefs among each agent's network contacts.

For random networks, the difference between global and local updating of trust can make a significant difference in observed polarization. Figure 2 shows a typical evolution with $\tau = .5$ on a global scale. Figure 3 shows a comparative evolution with $\tau = .5$ on a local scale. Figure 4 generalizes the results for τ between 0 and .7 with histograms for belief distributions in the case of global updating (left) and local updating (right).

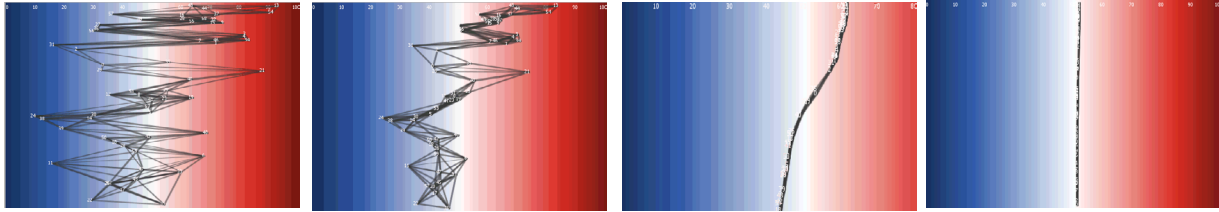


Fig. 2. Opinion convergence given linear trust updating on global scale. Generations 5, 15, 25 and 30 shown.

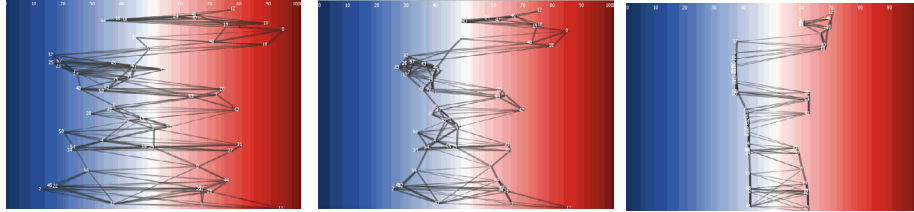


Fig. 3. Opinion convergence given trust updating on local distance. Generations 5, 15, and 30 shown.

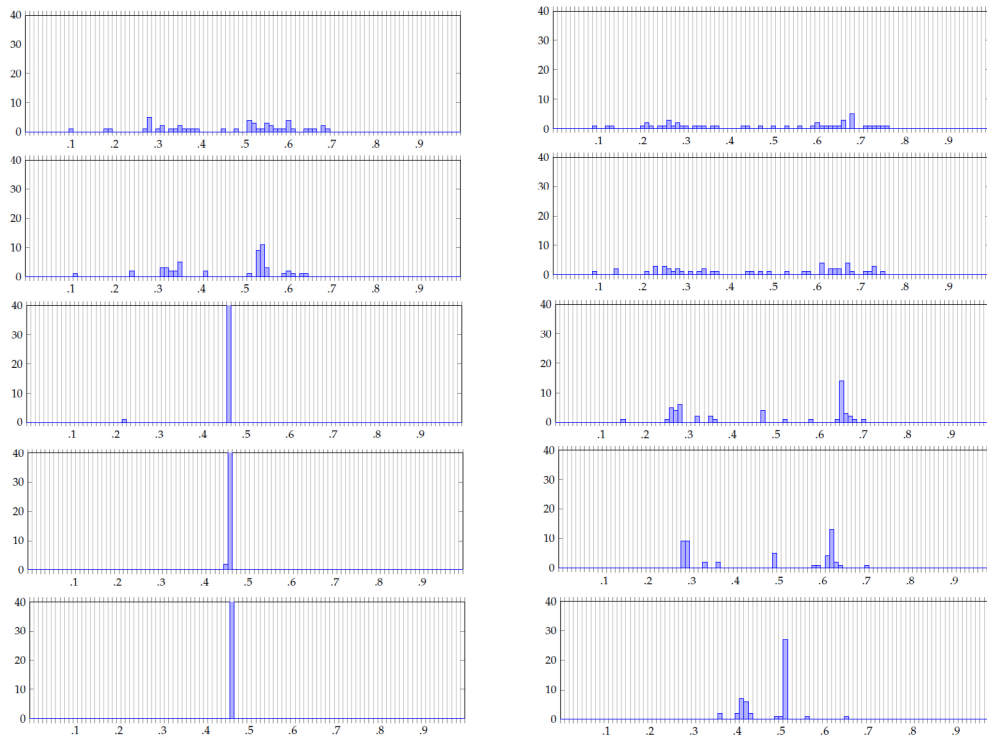


Fig. 4 Histograms for belief distributions with global trust updating (left) and local (right) for τ values of .05, .1, .2, .3, and .4.

The difference in global and local scaling of a linear trust updating function clearly does make a difference in the emergence of polarization. There certainly are other important factors, however. Other aspects of our work, not presented here, indicate that although media sources do not dampen the effect, increased population size and the shift from random to scale-free and spatial networks can be of importance.

References

- Brownstein, R. (2007). *The Second Civil War: How Extreme Partisanship has Paralyzed Washington and Polarized America*. New York: Penguin.
- Fiorina, M., Abrams, S. J., and Pope, Jeremy C. (2010). *Culture Wars? The Myth of a Polarized America*. New York: Longmans.
- Lord, C. G., Ross, L., and Lepper, M. R. (1979). Biased assimilation and attitude polarization: The effects of prior theories on subsequently considered evidence. *Journal of Personality and Social Psychology* 37, 2098-2109.
- Visser, P. S., & Cooper, J. (2003). Attitude Change. In M. Hogg and J. Cooper, eds., *Sage Handbook of Social Psychology*. London: Sage Publications.

Evolution and emergence of sign production and interpretation

Angelo Loula^{1*}, Ricardo Gudwin² and João Queiroz³

¹Cognitive and Intelligent Systems Lab (LASIC), State University of Feira de Santana, Bahia, Brazil

²Department of Computer Engineering and Industrial Automation, School of Electrical and Computer Engineering, University of Campinas (UNICAMP), Brazil

³Institute of Arts and Design, Federal University of Juiz de Fora (UFJF), Brazil

* angelocl@ecom.ufes.br

Extended Abstract

Computational modeling of the emergence of semiotic processes, such as language and communication, has been consolidating as an important methodology (Wagner et al., 2003; Noble et al., 2010). As the main form of interaction between agents in these experiments, communication has been a significant research subject. Primarily, it depends on the production of representations (by an utterer) and the interpretation of them (by an interpreter). Despite the fact that representation processes are in the foundations of communication, little discussion about such processes can be found, such as, the emergence of fundamental types of representations and their referential relations.

We have previously simulated the emergence of interpretation of two different types of representations (symbols and indexes) in communicative interactions (Loula et al., 2010), and studied further the cognitive conditions to such processes (Loula et al., 2011). Here we propose to evaluate representation processes in the emergence of both interpretation and production of multiple representations, with multiple referents. To do so, we apply a neural network model as the cognitive architecture for creatures, which can become utterers and interpreters. The experiment applies C.S. Peirce's pragmatic theory of signs as theoretical basis.

To test the conditions for the emergence of semiotic processes, artificial creatures are evolved to collect resources in a virtual environment. Two types of resources can be found in the environment, with positive and negative values, and creatures can vocalize two types of signs. Creatures are controlled by a feed-forward neural network with three layers. For better analysis of neural network activation patterns, we applied a winner-takes-all (WTA) mechanism to the middle layer and output layer. Auditory middle layer can be connected to the output layer (type 1), probably defining an indexical sign interpretation, or can be connected to the visual middle layer, defining an associative memory between auditory activations and visual activations (type 2), and defining a symbolic sign interpretation. Evolution allows the creatures to adapt to the task of collecting positive resources and avoiding negative resources.

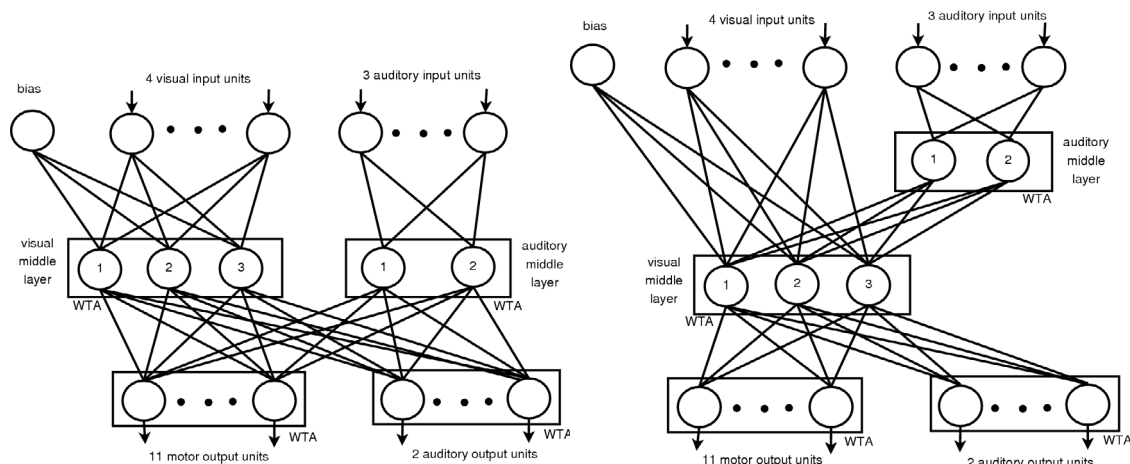


Figure 1: Neural networks used by creatures. Left: Type 1 architecture. Right: Type 2 architecture.

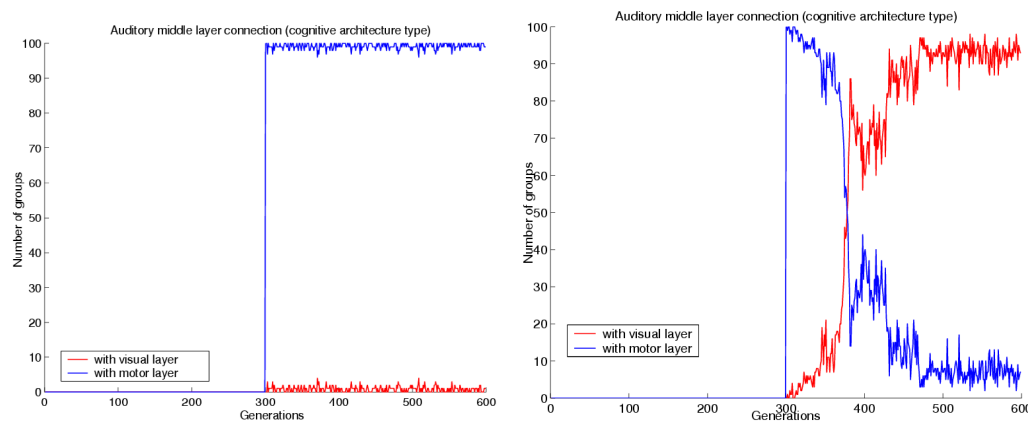


Figure 2: Evaluation of auditory middle layer connection for the first simulation (left) and for second simulation (right).

We ran two simulations of the experiment. In the first one, only the neuron with the highest positive activation has output of 1.0 (the others became null), and in the second simulation, the activation value of this neuron must be 1.0 higher than the second highest active neuron, therefore it is harder to learn motor coordination in this second configuration. In the first simulation, motor actions could be easily coordinated with sensorial input, and the adaptive behavior evolved was a direct response to the communicated signs, an indexical interpretation (figure 2). Increasing the cost of cognitive traits acquisition in the second simulation, symbolic interpretation of signs was the adaptive response (figure 2). The proposed neural network allowed a detailed inner observation of cognitive processes during experiments and therefore to analyze the semiotic relations being established in the utterer and in the interpreter.

References

- Loula, A., Gudwin, R. and Queiroz, J. (2011) Cognitive conditions to the emergence of sign interpretation in artificial creatures. In *Proceedings of the 11th European Conference of Artificial Life, ECAL'11*, 2011, France. (p. 497-504)
- Loula, A., Gudwin, R., and Queiroz, J. (2010) On the emergence of indexical and symbolic interpretation in artificial creatures, or What is this I hear? In Fellermann, H., et al., editors, *Artificial Life XII*, pages 862–868. MIT Press.
- Noble, J., Ruiter, J. D., and Arnold, K. (2010). From monkey alarm calls to human language: How simulations can fill the gap. *Adaptive Behavior*, 18(1), 66–82.
- Wagner, K., Reggia, J. A., Uriagereka, J., and Wilkinson, G. S. (2003). Progress in the simulation of emergent communication and language. *Adaptive Behavior*, 11(1), 37–69.

Herding Behaviour Experimental Testing in Laboratory Artificial Stock Market Settings. Behavioural Foundations of Stylised Facts of Financial Returns.

Viktor Manahov* and Robert Hudson

Newcastle University Business School

[*v.manahov@newcastle.ac.uk](mailto:v.manahov@newcastle.ac.uk)

Extended Abstract

Many scholars express concerns that herding behaviour causes excess volatility, destabilises financial markets and increases the likelihood of systematic risk. The growth of investment institutions over the years has increased the possibility of herding. For example, the percentage of the UK stock market held by individuals dramatically decreased from 54% in 1963 to 12.8% in 2006 [Hudson and Atanasova (2009)]. Herding behaviour is more likely to occur in markets dominated by institutions because managers employed by institutions operate in the market to make money and retain their jobs. Their performance is often based on large compensation packages. The intuition behind this claim is that the profit condition-particularly a mandate to achieve a minimum benchmark return could lead to weaker incentives for individuals to deviate from the benchmark and hence effectively reduces the competition among them. The lack of competition may lead to the convergence of opinions and the adoption of similar investment strategies. Hence, herding behaviour is encouraged causing potential long-term market reverses and relaxed risk-management controls [Gompers and Metrick (2001), Wermers (1999), Scharfstein and Stein (1990)].

We use genetic programming (GP) software to evolve a stock market divided into two groups- a small subset of artificial agents called 'Best Agents' and a main cohort of agents named 'All Agents'. The 'Best Agents' perform best in terms of the trailing return of a wealth moving average. 'All Agents' represent the remainder of the virtual market population. We then investigate whether herding behaviour can arise when agents trade in three separate artificial stock markets based on an index and two securities- the Dow Jones, general Electric and IBM. Our research uses real historical quotes of the three financial instruments to analyse the behavioural foundations of stylised facts such as Leptokurtosis, non-IIDness and volatility clustering.

Our experimental results show that an artificial stock market populated by a small subset of best performing agents behaves differently from a market with greater genetic diversity. Although there is no discernible difference in terms of volatility, the market based on the behaviour of 'All Agents' exhibit less herding and is more efficient than the segmented market populated by 'Best Agents' (Figure 1 and Figure 2). Hence, the price formation process caused by the collective behaviour (competition and co-evolution) of the entire market is a better predictor than any small fraction of agents. This is a result of the greater genetic diversity that is presented in the total population. Enhanced diversity means more heterogeneous trading rules and behaviour leading to greater flexibility in the virtual market clearing price mechanism. In this particular case we find no support for the Marginal Trader Hypothesis which holds that a small group of traders such as 'Best Agents' keep an asset's market price equal to its fundamental value and steer markets to efficient levels. Moreover, in line with previous research, there is some evidence of more herding in a group of stocks such as the Dow Jones index than in individual stocks like General Electric and IBM. However, our empirical findings suggest that the magnitude of herding is far from dramatic and does not exhibit long-run mispricing of assets and bubble formation.

Greater genetic diversity also means less non-linear dependence, more unpredictability and therefore an enhanced level of randomness in the return series. Hence, these series can be considered more efficient. Unlike small groups of artificial agents where substantial volatility clustering persists, the presence of more agents has led the market to lower levels of localised bursts in the amplitude of price fluctuations.

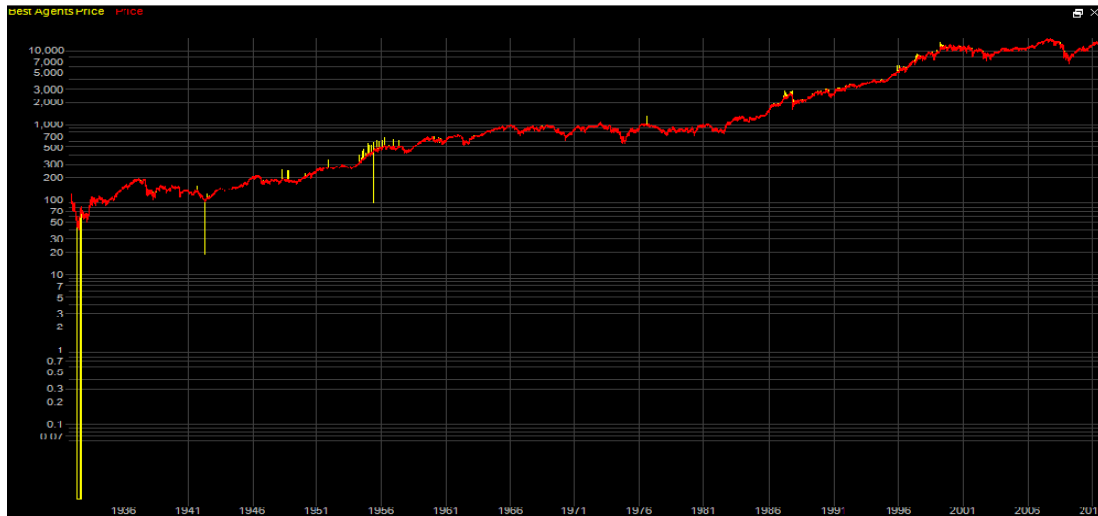


Figure 1. Significant herding behaviour observed in the Dow Jones time series (01/09/1931-17/06/2011) generated by 20% 'Best Agents' group size (2,000 artificial traders).

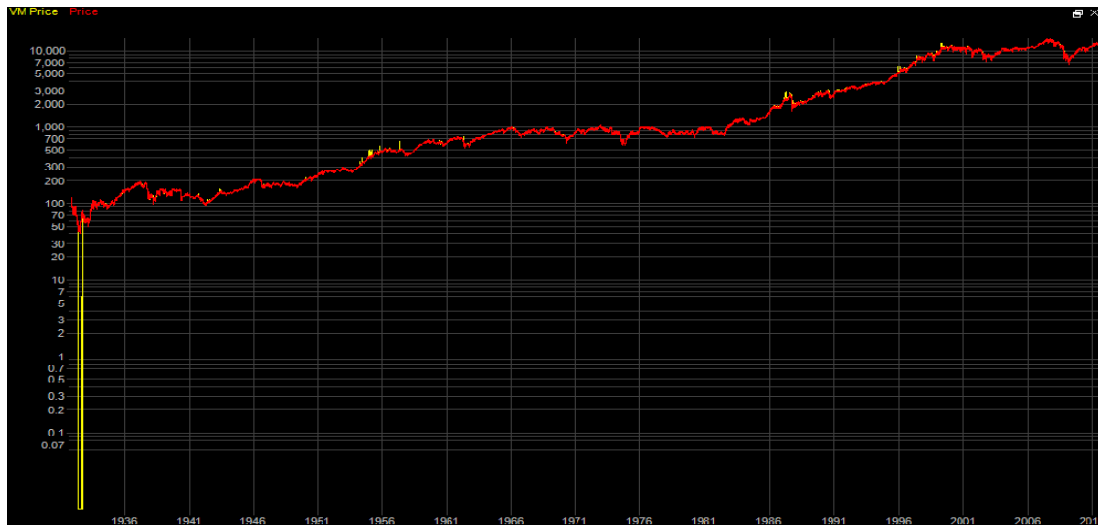


Figure 2. Insignificant herding behaviour observed in time series of the Dow Jones (01/09/1931-17/06/2011) generated by the reminder of the market (8,000 artificial traders).

References

- Gompers, P. And Matrick, A. (2001). Institutional Investors and Equity Prices. 116 *Quarterly Journal of Economics* 229.
- Hudson, R. And Atanasova, C. (2009). Equity Returns at the end of the month: Further Confirmation and Insights. *Financial Analysts Journal*, 65(4), pp.14-16.
- Scharfstein, D., and Stein, J. (1990) Herd Behaviour and Investment. *American Economic Review*, Vol.80, pp.465-79.
- Wermers, R. (1999). Mutual Fund Herding and the Impact on Stock Prices. 54 *Journal of Finance* 58.

An Algorithm to Create Phenotype-Fitness Maps

Jean-Baptiste Mouret^{1,2}, Jeff Clune³

¹Université Pierre et Marie Curie-Paris 6, UMR 7222, ISIR, France

²CNRS UMR 7222, ISIR, France

³Cornell University, Ithaca, USA

mouret@isir.upmc.fr

Extended Abstract

Understanding the relationships between phenotypic characteristics and fitness is central to evolutionary biology and the design of new evolutionary algorithms (EAs). Whether in computational models of evolution (Kauffman, 1993; Adami, 1998; Lenski et al., 2003) or in evolutionary algorithms (Goldberg, 1989), the common approach is to perform selection based on fitness and study the phenotypes that evolve. Unfortunately, computational evolution tends to be highly convergent, meaning there is little diversity in the population and thus little variation along key phenotypic dimensions. Such a lack of diversity prevents an understanding of how fitness would change along those dimensions ‘had evolution searched there’. The problem is compounded by the fact that fitness landscapes often have many local optima that populations get stuck on, which makes it difficult to know if there are higher fitness peaks in other areas of the fitness landscape that evolution failed to discover. Both biologists and engineers often spend a lot of time asking that very question, and would benefit from tools that help them answer it.

Here we introduce an algorithm to compute *phenotype-fitness maps* as a way to understand the relationship between phenotypic dimensions and fitness. The central idea is to explicitly select for fit organisms in *all areas* of a phenotype landscape, where the axes of that landscape are defined by phenotypic dimensions of interest. To produce such maps, we introduce the Multi-Objective Landscape Exploration (MOLE) algorithm, which is a multi-objective evolutionary algorithm, specifically NSGA-II (Deb, 2001), with two objectives: (1) searching for new organisms that are far from solutions already generated, with distance measured in a Cartesian space defined by the key dimensions, and (2) generating highly fit organisms. With MOLE, scientists can see how fitness changes as a function of various phenotypic dimensions (Figure 1). This combination of a fitness objective and an archive-based exploration objective is similar to “novelty-based multi-objectivization” (Mouret, 2011; Lehman and Stanley, 2011).

We investigate phenotype-fitness maps produced via MOLE by evolving the topology and parameters of feed-forward neural networks to recognize binary patterns in an 8-pixel retina (Kashtan and Alon, 2005). Fitness is the normalized error for all 256 possible input patterns. Two encodings are investigated: a direct encoding that allows for arbitrary, non-recurrent topologies (DNN, see Mouret and Doncieux (2012)), which is similar to NEAT (Stanley and Miikkulainen, 2002), and a more constrained direct encoding that is feedforward and specifies the number of layers and maximum number of neurons per layer (KA, see Kashtan and Alon (2005)). These constraints reduce the search space to a region known to contain perfect solutions. Each phenotype landscape shows the highest-performing organism at each location found during 30 independent evolutionary runs. As a control, we also conducted 30 runs per encoding with a typical EA, represented by NSGA-II with a single fitness objective.

Preliminary results (Figures 1 and 2) show that generated phenotype-fitness maps can provide an informative window into how phenotypic dimensions relate to fitness. Moreover, a single MOLE run can find high-performing organisms with a variety of phenotypic traits instead of the homogenous set typically generated by a single EA run: The quality and diversity of solutions MOLE generates suggests that it could also represent a powerful alternative to classic, convergent EAs.

References

- Adami, C. (1998). *Introduction to artificial life*, volume 1. Telos Pr.
- Claussen, J. C. (2007). Offdiagonal complexity: A computationally quick complexity measure for graphs and networks. *Physica A: Statistical Mechanics and its Applications*, 375(1):365–373.
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Wiley.
- Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley.
- Kashtan, N. and Alon, U. (2005). Spontaneous evolution of modularity and network motifs. *PNAS*, 102(39):13773–13778.
- Kauffman, S. A. (1993). *The origins of order: Self organization and selection in evolution*. Oxford University Press, USA.

This research was funded by the ANR (project ANR-09-EMER-005-01) and an NSF Postdoctoral Research Fellowship to JC (DBI-1003220).

- Kim, J. and Wilhelm, T. (2008). What is a complex graph? *Physica A: Statistical Mechanics and its Applications*, 387(11):2637–2652.
- Lehman, J. and Stanley, K. (2011). Evolving a diversity of creatures through novelty search and local competition. In *Proc. of GECCO*.
- Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423(6936):139–144.
- Li, M. and Vitányi, P. (2008). *An introduction to Kolmogorov complexity and its applications*. Springer-Verlag New York Inc.
- Mouret, J.-B. (2011). Novelty-based multiobjectivization. In *New Horizons in Evolutionary Robotics: Extended Contributions from the 2009 EvoDeRob Workshop*, pages 139–154. Springer.
- Mouret, J.-B. and Doncieux, S. (2012). Encouraging behavioral diversity in evolutionary robotics: an empirical study. *Evolutionary Computation*, 20(1):91–133.
- Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.

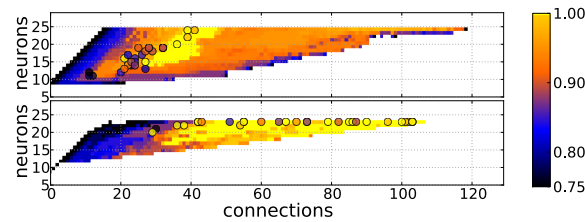


Figure 1: **Experiment:** A straightforward set of dimensions for evolving neural networks: number of nodes vs. number of connections. Fitness is colored. Circles indicate the best solution from each of 30 standard EA runs (some overlap). **Results:** (Top) Phenotype-fitness map obtained with the DNN encoding. The MOLE algorithm found 98 distinct perfect solutions (bright yellow areas) whereas 30 runs of a standard EA found only 6 perfect solutions (bright yellow circles). (Bottom) Phenotype-fitness map obtained with the KA encoding. The MOLE algorithm found 221 perfect solutions whereas a standard EA found 15. **Comments:** These maps reveal relationships between the dimensions and fitness, such as the minimum number of neurons and connections needed to solve the problem. The maps provide more insight into these relationships than the EA does alone. The maps also reveal the impact of different encodings: while all KA networks are expressible in the DNN encoding, evolution with DNN is less likely to evolve fit solutions with many connections, shedding light on why the constraints in the KA encoding improve its performance.

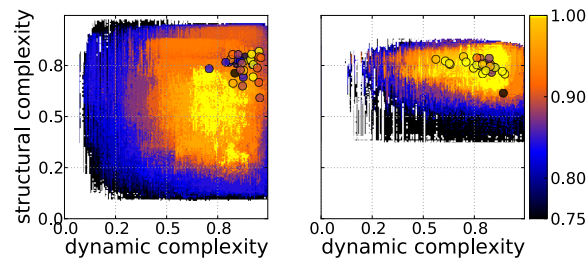


Figure 2: **Experiment:** We tested a second set of phenotypic dimensions on the same problem to understand the relation between fitness and *network complexity*. Of the many ways to evaluate structural complexity in networks (Kim and Wilhelm, 2008), we chose “off diagonal complexity” (Claussen, 2007) because it is fast to compute. With this measure, the complexity of fully connected networks and completely regular networks is zero, the complexity of a random graph is intermediate, and the highest complexity scores correspond to scale-free networks and hierarchical trees. In addition to structural complexity, neural networks can vary in the dynamics of their activity. This complexity can be captured by computing the Kolmogorov complexity of the sequence of outputs of each neuron for each input pattern. Here we approximate Kolmogorov complexity using the gzip2 compressor (Li and Vitányi, 2008). **Results:** (axes are normalized) (left) DNN phenotype-fitness map: MOLE found 2536 distinct perfect solutions whereas a standard EA found 6 (bright yellow circles) (right) KA encoding phenotype-fitness map: MOLE found 1690 perfect solutions whereas a standard EA found 15. The maps illuminate relationships between the dimensions, although the illumination is not flawless, as some perfect solutions found by the standard EA were not located by MOLE.

Multi-Robot, Multi-Patch Foraging with Maximum Sustainable Yield

Zhao Song and Richard T. Vaughan

School of Computing Science, Simon Fraser University, British Columbia, Canada
 {zhaos,vaughan}@sfu.ca

Extended Abstract

Introduction

We introduce the Sustainable Robot Foraging (SuRF) problem, in which one or more robots must maximize the long-term profit obtained by harvesting resources from the environment. When the reward per unit harvested is constant or only slightly discounted over time this implies that the sources of resource must never be destroyed by over-harvesting, while under-harvesting fails to maximally exploit resources. This is a fundamental problem for living or artificial systems that aim to exploit biomass resources for long periods. The availability of resources over time is modeled using the classical logistic function originally proposed by Verhulst (1838) to model population growth, and since applied to the growth of tumors and many other natural systems. The logistic model improved on the earlier exponential growth model of Malthus (1798) by recognizing that populations generally can not grow unbounded, with growth limited as resources consumed by the existing population become scarce. A formula for obtaining the optimal harvest rate in systems with logistic growth was first described by Hjort et al. (1933) in their study of maximizing fish catches, and became well known in this context as the Maximum Sustainable Yield.

To apply these insights to the robotics context, we investigate a foraging problem in which autonomous robots must collect *pucks*; generic atomic objects of value to the robots' owner. Pucks are not distributed at random in the environment, but exist in areas of locally high density called *patches*. The number of pucks in a patch (the *patch size*) changes over time according to the logistic function, simulating a naturally regrowing resource that is harvestable in discrete units, such as mushrooms, acorns, fruits, animals and fish.

Once collected, pucks must be delivered to a central collecting point, at which time the robot system is credited with one unit of reward. Our goal is to maximize the total reward obtained by the system. If the reward per unit of resource is constant or discounted only slightly over time, then the optimal policy is to permanently sustain foraging while maximizing the instantaneous reward rate (Stephens et al., 2007;



Figure 1: Robots forage for resources that demonstrate logistic population growth. To obtain maximum sustainable profit, the robots must harvest resources at the rate that maximizes the rate of regrowth. This is the Maximum Sustainable Yield. [Artwork © Christine Larson]

Wawerla and Vaughan, 2010). To achieve this, robots must harvest resources from each patch at the rate that provides the fastest resource growth rate at that patch. This implies that the patch will remain at some ideal population size. Collect pucks too slowly or too quickly and the patch is less than optimally productive. If a patch size gets below some lower bound, it can not regenerate and is permanently destroyed.

We used the Maximum Sustainable Yield formulation to find the optimal robot work allocations for our robot foraging problem. Realizing the model in a numerical simulation, we observe a well-known problem with MSY: the system is dynamically sensitive to small perturbations, so that the fixed allocation does not provide good sustainable foraging. To cope with this we devised a simple feedback controller that locally modulates the foraging rate at each patch to achieve sustainability and close to optimal performance. We demonstrated the controller achieving optimal foraging in a simple robot simulator.

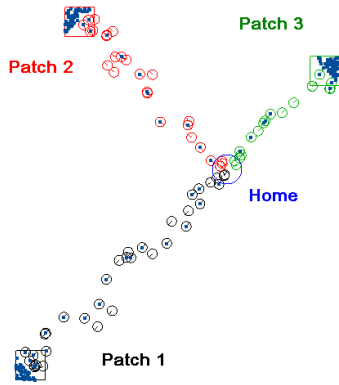


Figure 2: Screenshot from the Antix simulator. 80 robots (small circles) adaptively forage pucks (dark dots) from 3 patches (squares) and deliver them to the home (large circle).

Demonstration

We demonstrate a simple adaptive controller in the freely-available sensor-based robot simulator Antix¹. Pucks are placed at random in the patches, the robot drives between home and goal using a simple kinematic controller, and detects pucks using an on-board sensor with limited range.

There are three patches, all with the same logistic growth parameters, but located at different distances from home, at 2, 3 and 4 times unit distance, as shown in Figure 2. The overall performance metric we mean to optimize is the sustained delivery rate of the entire robot system, which is simply the total number of pucks delivered by all robots per unit time. The optimal delivery rate is 10 pucks unit time.

Figure 3 shows an example system evolution for 100 robots. Patch population plot (a) shows the robots initially over-harvest all patches and the populations drop quickly. Adapting to the falling population, the robots increase their sleep time and the population climbs again, overshooting the ideal size until the robots adapt again, bringing the population back to the approximately optimal size. The patch growth rate is shown in (b), and the puck delivery rate is seen in (c), climbing from zero as robots are deployed, rising above 10 pucks per unit time as the patch is over-harvested, dropping as the population declines, then converging close to the around the Maximal Sustainable Yield of 10 pucks per unit time for each patch. The excess work capacity has been turned into inactive robot “sleep” time to avoid over-harvesting.

Contributions

1. The introduction of the Sustainable Robot Foraging (SuRF) problem, and the first demonstration of sustainable robot foraging. This is the first work to examine optimal foraging strategies in robot systems where the robots’

¹<http://github.com/rtv/Antix>

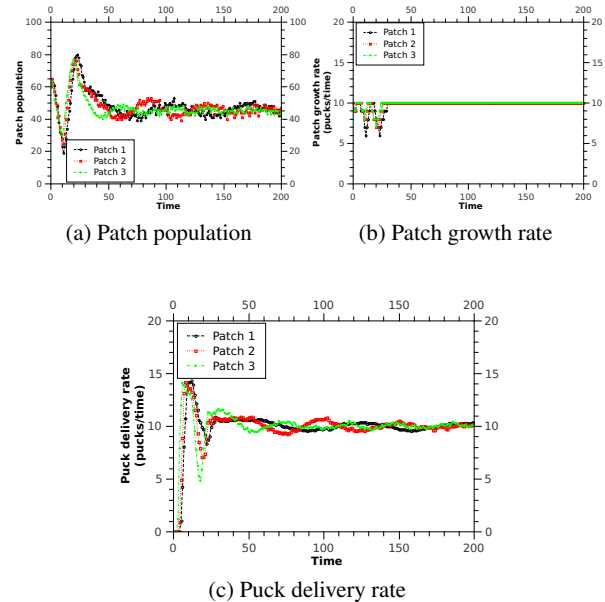


Figure 3: Results: 100 robots adaptively foraging 3 patches of randomly placed pucks. Time units are hundreds of simulated seconds. Allocation is patch 1:=43 robots, 2:=33, 3:=24. Sustainable and optimal (maximum productivity is 10 pucks/unit time).

activity feeds back into the subsequent productivity of the environment;

2. The first application of Maximum Sustainable Yield model to robot foraging.

This is an early step towards the development of machines that can harvest biomass from the environment indefinitely without damaging it. This is a challenge that has defeated even the smartest primates, historically.

References

- Hjort, J., Jahn, G., and Ottestad, P. (1933). The optimum catch. *Hvalradets Skrifter*, 7:92–127.
- Malthus, T. (1798). An essay on the principle of population. *London*.
- Stephens, D. W., Brown, J. S., and Ydenberg, R. C. (2007). *Foraging*. University of Chicago Press, Chicago.
- Verhulst, P. (1838). Notice sur la loi que la population suit dans son accroissement. *Corr. Math. et Phys*, 10:113.
- Wawerla, J. and Vaughan, R. T. (2010). Online robot task switching under diminishing returns. In *Proceeding of the Twelfth International Conference on Artificial Life (ALife XII)*, pages 789–796.

On-line, On-board Evolution of Reaction-Diffusion Control for Self-Adaptation

Jürgen Stradner, Heiko Hamann, Payam Zahadat, Thomas Schmickl, and Karl Crailsheim

Artificial Life Lab of the Department of Zoology, Karl-Franzens University Graz, Universitätsplatz 2, A-8010 Graz, Austria
{juergen.stradner, heiko.hamann, payam.zahadat, thomas.schmickl, karl.crailsheim}@uni-graz.at

Extended Abstract

Introduction

Evolutionary robotics (ER) aims to automatically develop sensor-actuator control of agents by using techniques of artificial evolution. The first review article in the field was published in the mid nineties by Mataric and Cliff (1996). In their article they are clear about a list of fundamental problems arising when using ER in simulation or in the real world. Unfortunately, 16 years after tackling the problems concisely, their hope expressed in the very last sentence of the paper did not fully become true yet: “If the challenges can be successfully addressed, the use of evolutionary techniques may become a viable alternative to manual design.” Several of these challenges were, however, treated in-depth. Concerning the methodology of using a simulation environment to develop a controller followed by transferring it to the robot, the reality gap problem was addressed. One of the methods of how to bridge the reality gap was proposed by Jakobi (1997). The idea is to mask non relevant aspects of the environment by noise. When using real robot hardware the problem of keeping the robots’ autonomy for long periods arises. A technical solution for energy autonomy was proposed by Watson et al. (2002) in the form of electrified floors. Obviously this method is not applicable outside of the lab. However, an environment with unforeseen conditions is meant to be the actual operation ground of ER in the first place. A broader overview of methods which address the challenges reported by Mataric and Cliff (1996) can be found, for example, in Nolfi and Floreano (2000).

One of the major goals of the projects SYMBRION and REPLICATOR is to create controllers for modular robots which autonomously dock to form robot organisms. Due to the combinatorial explosion of possible robot organism configurations the robot controllers cannot be pre-defined for particular organism shapes. Therefore, we are constrained to an approach of ER, which proved to be a promising method in the last years: on-line evolution, whereas we rely on the definitions of this category established by Watson et al. (2002) and Eiben et al. (in Levi and Kernbach (2010), ch. 5.2). The idea is to tune the controller of the robot while it is actively trying to

achieve the given objective. Thus, environmental changes or changes in the task are immediately incorporated into the adaptation process of the robot controller.

The method of on-line, on-board ER is combined with a bio-inspired controller called AHHS (Artificial Homeostatic Hormone Controller), see Schmickl et al. (2011); Hamann et al. (2012). AHHS was designed for high evolvability in multi-modular robotics. To our knowledge, the presented results represent the first investigations of using a reaction-diffusion based robot controller in an on-line, on-board experiment.

We report experiments using both a real robot and a simulator which represents the applied robot in detail. The used simulation environment is Robot3D and the used hardware platform is the ‘backbone robot’ (Levi and Kernbach, 2010) from our projects. The number of evaluations was reduced from 2500 in simulation to 600 in the robot experiments.

The complexity of the investigated task is limited because this is a first case study of our on-line, on-board evolution approach using AHHS on hardware. By restricting the actuator control values we limit the robot’s DOF to one: motion back and forth between two walls. We use a single input s from a front proximity sensor which is scaled to $s \in [0, 1]$. The fitness function F of the task is

$$F(s) = \min(2s, -2s + 2), \quad (1)$$

where s is the sensor value at the end of the evaluation phase. Maximal fitness $F = 1$ is achieved for $s = 0.5$ while other values are linearly scaled to $F(s) = 0$ for $s \in \{0, 1\}$. The robot has to position itself such that medium sensor readings are obtained. One advantage of this simple task is that we are able to reflect it well in simulation especially concerning the sensor input. A difficulty of this task is that an on-line measured fitness is maximally correlated with the initial position and, hence, can be different from the actual fitness that is determined in the post-evaluation based on several initial conditions. Hence, intensive post-evaluations were done by rerunning all individuals of the last 50 evaluations for 21 initial positions distributed over the whole space. The average of these 21 tests is the post-evaluated fitness.

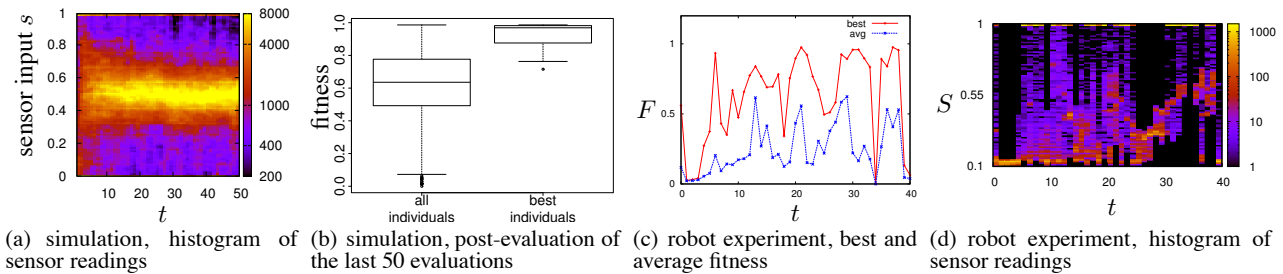


Figure 1: Results in simulation ($n = 12$, $(\Delta t = 1) \hat{=} 50$ evaluations) and robot experiment ($n = 1$, $(\Delta t = 1) \hat{=} 15$ evaluations).

Results

In simulation we have done $n = 12$ evolutionary runs with 2500 evaluations each. The summed sensor values over all evaluations of all runs are shown in a histogram in Fig. 1(a). An increment of 1 on the t -axis represents 50 evaluations. It can be seen that already at $t = 5$, hence after approximately 250 evaluations, a majority of sensor values is close to the optimum of $s = 0.5$. Still, exploration of the search space is not stopped until the very end of the experiment as one can tell from the filled bins all over the diagram (robots keep moving through the arena). Even at $t = 50$ there are still some sensor values of $s = 0$ and $s = 1$ which show that the robot was sometimes situated close to the front and back wall. The results of the post-evaluation are shown in Fig. 1(b). The left boxplot shows the data for all $12 \times 50 = 600$ controllers, the right one gives only the best controller of the last 50 evaluations for each of the 12 runs. For comparison note that a trivial but good behavior is to not move which would result in a post-evaluated fitness of 0.46. The median over all controllers (0.64) is clearly above that. The fitness of the best controllers (> 0.71) indicates close to perfectly adaptive behavior.

For experiments on real robot hardware we report a small case study. A video of this run is available on-line¹. Fig. 1(c) is a typical representative of on-line evolution runs. The average fitness initially increases slowly but both best and average fitness are mostly characterized by severe jumps. Fig. 1(d) reveals some causes of the fitness jumps. In the first 60 evaluations the robot positions itself at the back wall. For a long time ($4 < t < 33$) the evaluated behaviors seem to keep a reasonable balance between forward- and backward-moving until the robot places itself at the front wall ($t = 33$). At that time no backward-moving behavior seems to be available in the population of controllers. Hence, the robot stays there for more than 15 evaluations. This reveals a major problem of on-line evolution. It is the tradeoff of keeping a balance between exploration and exploitation. A drawback of exploration is the peril of evaluating controllers that might destroy a good initial condition. A drawback of exploitation is to forget behaviors that might

not be helpful right now but might help in other situations that occur later. Still, we conclude that we were able to optimize an AHHS in on-line, on-board evolution on a robot. The robot clearly performed better than random.

Discussion

This article describes experiments in ER using on-line evolution in combination with a hormone-based controller. We did our experiments on two different platforms: a simulation environment providing a detailed representation of the robot, and the real robot hardware. In this work we avoided some challenges of ER, for example, the reality-gap problem because we started the experiments on the hardware initialized with random controllers. Instead of transferring pre-evolved controllers to the robot for further evolution, we transferred just the knowledge of EA-parametrization from the simulation. In our ongoing and future research we continue this approach and focus on achieving fully self-adaptive robot systems based on artificial evolution.

Acknowledgments This reported research is supported by: EU-IST-FET project ‘SYMBRION’, no. 216342; EU-ICT project ‘REPLICATOR’, no. 216240; Austrian Federal Ministry of Science and Research (BM.W.F.).

References

- Hamann, H., Schmickl, T., and Crailsheim, K. (2012). A hormone-based controller for evaluation-minimal evolution in decentrally controlled systems. *Artificial Life*, 18(2):165–198.
- Jakobi, N. (1997). Evolutionary robotics and the radical envelope-of-noise hypothesis. *Adaptive Behavior*, 6(2):325–368.
- Levi, P. and Kernbach, S., editors (2010). *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution*. Springer.
- Matarić, M. J. and Cliff, D. (1996). Challenges in evolving controllers for physical robots. *Robotics and Autonomous Systems*, 19(1):67–83.
- Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press.
- Schmickl, T., Hamann, H., and Crailsheim, K. (2011). Modelling a hormone-inspired controller for individual- and multi-modular robotic systems. *MCMDS*, 17(3):221–242.
- Watson, R., Ficici, S., and Pollack, J. (2002). Embodied evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39(1):1–18.

¹<http://youtu.be/P4w3ijRjUy0>

Challenges for A-Life Approach to Artificial Cognition: in Search for Hierarchy of Cognitive Systems

Borys Wróbel^{1,2,3}

¹Evolutionary Systems Laboratory, Adam Mickiewicz University in Poznań, Poland

²Systems Modeling Laboratory, IO PAN, Sopot, Poland

³Institute for Neuroinformatics, University of Zurich & ETH Zurich, Switzerland

wrobel@evosys.org

Extended Abstract

Building artificial cognitive systems is a multidisciplinary endeavor, but the diversity of backgrounds and motivations make it difficult to formulate common objectives and challenges. Many unnecessary arguments can be avoided by clearly stating the motivations of a particular research program. The common sets of motivations in the field are sometimes conflicting. We think three sets can be identified: (i) engineering systems which have capabilities equivalent to biological systems, but which are based on different principles, (ii) engineering systems that use solutions that are close to biological materials and/or are bio-compatible, (iii) building biologically-inspired systems in order to inform biology, to understand-by-construction, or to test biological hypotheses. The last set may be based on a view that a true understanding amounts to the ability to reverse-engineer. However, not all features of biological systems may be relevant for cognition *per se*. For example, we recognize that (self-)construction, regeneration, reproduction, and evolution may be related and necessary (but not sufficient) for bio-cognition (Duijn et al. 2006), but perhaps not for artificial cognition. In other words, we consider these features to be aspects of construction that should be differentiated from the operational issues (Beer, 1997), although we do recognize that a system that lacks these features may not be able to carry the cognitive functions in a changing environment over a long time. We share the view that cognition can be cast as sensorimotor coordination and requires embodiment (Duijn et al. 2006). To sum up, in our opinion, autopoiesis is not sufficient nor necessary for cognition, and that viability constraints can be external, but not arbitrary when viewed from the evolutionary perspective (in which the individuals that do not meet the constraints do not leave progeny; cf. Bourguine and Stewart, 2004; Froese and Ziemke, 2009).

Our objective here is to create a list of landmarks on the road towards artificial cognition that can be phrased in terms of incremental improvements, that would be welcoming evolutionary and developmental approaches (although not exclusively), and that would allow, on one hand, mapping existing A-Life approaches, and on the other hand, well-known biological systems. We would like the landmarks to be formulated in very practical terms, so that an implementation of an agent and its environment (Beer, 1997) can be easily imagined, and in a way that is compatible with different sets of motivations. We would also like to cover the whole spectrum of cognitive behaviors. Relating the landmarks to biology can provide an intuition for implementation, but also a way to benchmark the achievements and to approximate the difficulty of particular challenges (cognitive abilities observed in many independent, ancient lineages, suggest that the related task is easy).

We recognize the fact that increased computational abilities and memory capacity can allow for increased precision, for dealing with larger-scale problems, and for dealing with more uncertainty/noise/distractors. Another dimension to a particular challenge can be added by considering the rate of change of the physical world relative to the temporal scale relevant to an individual agent (requiring higher abilities for learning /adaptation/plasticity). However, in order for the list to be useful, it should not be long, and it is advisable to identify which landmarks require inherently different cognitive skills. The final structure of the challenges may be hierarchical should refinement (sub-challenges) be necessary, and it may not need be one-dimensional, could be partial, or indeed tree-like, but our first attempt for the list of challenges for cognitive systems (which specify the classes of systems that can meet them) is as follows:

1. *Sensing with feedback(s) to reach optima of environmental gradients.* Example: search/avoidance of objects that are sources of diffusive substances, agent placed at random position in the environment with random position of sources. The dimension of adaptation (this may mean choosing an appropriate action from a pre-specified repertoire of behaviors) may involve changes in the relevance of gradients at a time scale relevant to an individual agent (e.g., Izquierdo and Harvey, 2007; Joachimczak and Wróbel, 2010).
2. *Taking advantage of spatial/temporal structure of the environment.* This challenge requires an internal representation of the environment. Example: an agent able to learn the position of the objects in the environment or regularities of the neighborhood relations between objects and able to navigate on the map regardless of their initial position. Adaptation to the changes in the

regularities adds another dimension to this challenge. Still another facet of this challenge is the length of the temporal delay between the sensory information and the necessary action.

3. *Manipulating the external physical environment.* Example: an agent which stores objects belonging to different categories at different locations (biological motivation: internal storage limitations). Here also there is a possibility that the relevance of the objects to the agent may change.

4. *Taking advantage of the regularities in the behavior of other agents.* Example: agent of type A which takes advantage of the fact that another agent in its environment, B, in certain situations is able to find objects of interest to A; A can follow B should such a situation arise. In other words, A offloads to B sensing of different objects, all of interest to A, or to take advantage of resources accumulated by B either internally (predation) or externally (for example, if B stores objects as described for challenge 4). Meeting this challenge may require an internal model of other agents and their possible intentions/motives.

5. *Taking advantage of the knowledge about the cognitive abilities and limitations of other agents* in order to influence their behavior. For example, A would make B collect objects of interest to A. This challenge can be framed in an egoistic fashion or in a framework of cooperation (reciprocal altruism).

The challenges above highlight the importance of specifying the environment in which a particular class of behavior can be expressed (Trewavas, 2003), the way in which the complexity of the environment, diversity of other agents, novelty of tasks, time scales of regularities can be varied, and limitations on the agent perception and computational resources imposed (cf. Hernandez-Orallo and Dowe, 2010). We also think that the while each challenge by itself can be made more complex by requiring balancing short-term and long-term goals, less difficult challenges can be met, in general, at a shorter temporal scale. We stress again that this is work in progress. The examples have been formulated in terms of foraging for resources because it is an essential activity, seen in unicellular and multicellular organisms. At this point we have not attempted yet to link the classes above to biological organisms, but we note that rudimentary cognition (challenge 1 and 2) can be seen in animals with brains of about 1000 neurons, but also in unicellular organisms ("minimal cognition", Duijn et al. 2006), and plants (Trewavas, 2003).

It is important that the problems/challenges are formulated in such a way that would allow to test for the brittleness of solutions. The hope is that increasing the complexity of the tasks in the classes (along the added dimensions mentioned above) would require solutions which are not brittle, which involve internal representations, association of these internal symbols with the features of the external environment (symbol grounding or linking), manipulation of these symbols, and communication (this applies especially to challenge 5). We do not necessarily suggest that such communication would need to have features of human language or that internal representations would need to have features of human thought. Rather, we believe that in order to be efficient in a complex world, there is a need to cluster objects (or different forms of the signal about the same type of object that reaches the sensors in different environmental conditions), and to represent these clusters internally. The hope of the program outlined by the challenges listed above is that efficient inference about temporal (causal) and spatial relations between events or objects in a complex physical world may only be possible if symbols are manipulated internally. Manipulating them in a non-restricted but structured fashion may be necessary for internal redescription or reinterpretation and for efficient communication.

In summary, one could say that the whole program involves a two-fold **meta**-challenge. This meta-challenge consists firstly of identifying tasks, or aspects of tasks, specified precisely in terms of a platform/domain, which require internal representation (symbols, classifications) which are grounded (or linked to physical realities), and for which solutions based on acting in a purely reactive fashion or using simple signal processing are impossible or brittle. Secondly, the meta-challenge consists of finding an approach to assess if a particular solution indeed uses such internal representations or meanings.

Acknowledgments: The refinement of these ideas was made possible by the 2nd European Network for the Advancement of Artificial Cognitive Systems, Interaction and Robotics, and I am grateful to the members of the EUCogII/III community for discussion: Włodzimierz Duch, and especially Ricardo Sanz, Bill Sharpe, Aaron Sloman, Kostas Stathis, Ricardo Tellez, Serge Thill, Sandor Veres, and Georg von Wichert (members of group D during the "EUCogII Workshop on Challenges for Artificial Cognitive Systems II")

References

- Beer, R. D. (1997) The dynamics of adaptive behavior: a research program. *Robotics and Autonomous Systems*, 20: 257-289.
- Bourgine, P. and Stewart, J. (2004) Autopoiesis and cognition. *Artificial Life*, 10: 327-345.
- Froese, T. and Ziemke T. (2009) Enactive artificial intelligence: investigating the systemic organization of life and mind. *Artificial Intelligence* 173: 466-500.
- Hernandez-Orallo J. and Dowe D.L. (2010) Measuring universal intelligence: towards an anytime intelligence test. *Artificial Intelligence* 174: 1508-1539.
- Izquierdo, E. and Harvey I. (2007) The dynamics of associative learning in an evolved situated agent. In *Advances in Artificial Life: Proceedings of the 9th European Conference on Artificial Life*, pages 365-374. Springer-Verlag, Berlin, Germany.
- Joachimczak, M. and Wróbel, B. (2010a). Evolving gene regulatory networks for real time control of foraging behaviours. In *Artificial Life XII: Twelfth International Conference on the Simulation and Synthesis of Living Systems*, pages 348-355. MIT Press, Cambridge, MA.
- Trewavas, A. J. (2003) Aspects of plant intelligence. *Annals of Botany* 92: 1-20.
- van Duijn, M., Keijzer F., and Franken D. (2006). Principles of minimal cognition: casting cognition as sensorimotor coordination. *Adaptive Behavior*, 214: 157-170.

Synthetic Biology

Extended Abstracts

Protocellular energetics and autonomous functions

Anders N. Albertsen¹, Sarah E. Maurer¹, Jonathan Cape², Harold Fellermann¹, James M. Boncella², Hans-Joachim Ziock², Steen Rasmussen^{1,3} and Pierre-Alain Monnard¹

¹Center for Fundamental Living Technology (FLinT), University of Southern Denmark, Denmark

²Los Alamos National Laboratory, Los Alamos, NM 87545, USA

³Santa Fe Institute, Santa Fe NM, USA

monnard@sdu.dk

Extended Abstract

Living cells are in many respects the ultimate nanoscale chemical system. Within a very small volume they can produce highly specific and useful products by extracting resources and free energy from the environment. They are self-assembled, self-organized, highly energy efficient as well as capable of self-repair and self-replication.

Designing artificial chemical systems bottom up (artificial cells or protocells) endowed with these powerful capabilities is being intensively investigated by several teams, see Rasmussen *et al.*, 2009. Usually such chemical systems are based on a set of information and metabolic components encapsulated or co-located within the boundaries of self-assembled amphiphile structures (container). The protocell function is then supported by a continuous supply of substrates with high-energy bonds, e.g., NTPs, which will provide the energy needed to power the protocell metabolism. Thus, these systems lack the functionality of truly autonomous living systems because they do not convert primary energy, e.g., light or (geo)chemical energy, into new bonds, and by extension cannot create their building blocks from simple precursors, unless modern biological machinery is encapsulated in liposomes (Steinberg-Yfrach *et al.*, 1998).

Integrating energy conversion into a protocell function is not easily achieved, as the building blocks of protocellular systems are relatively complex from a chemical point of view. Using both simulations and wet chemistry, we have therefore attempted to realize this proposition by following two approaches: by creating simple, chemical systems (a) in which light energy is used to prepare protocell building blocks from relatively simple precursors and (b) which contain simple pigments that can harvest light and induce the formation of chemical gradients across amphiphile membranes, a first step towards self-sufficiency.

The first system (Fig. 1a) is based on the absorption of light by a photosensitive ruthenium complex that under the mediation of information molecules can transform amphiphile precursors into amphiphiles (Declue *et al.*, 2009; Maurer *et al.*, 2011) or induce ligation of nucleic acid oligomers (Cape *et al.*, 2012).

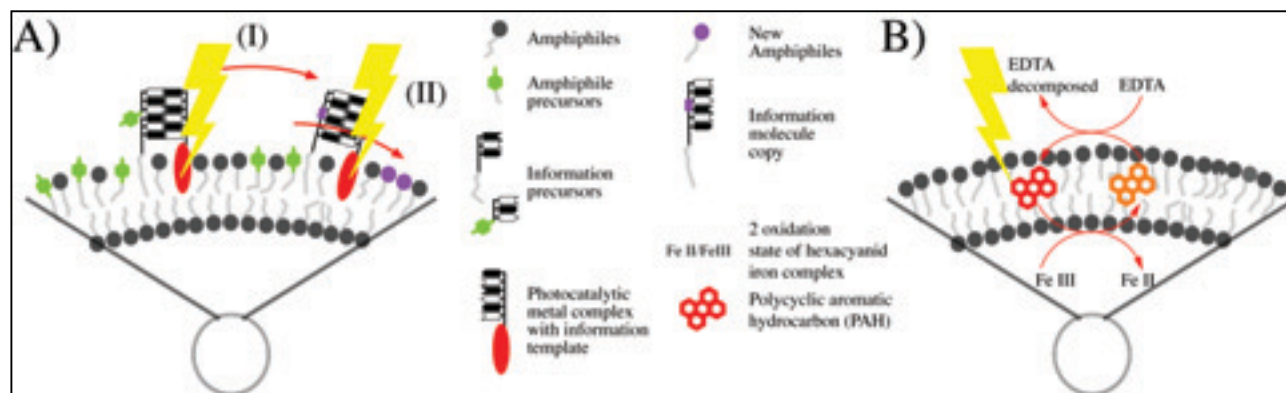


Figure 1: Light-harvesting systems and their supported reactions. A) A metal complex (ruthenium trisbipyridine) absorbs light and (A/I) initiates the replication of the protocell information and (A/II) converts precursor molecules into amphiphiles. B) PAHs (red) inserted in the membranes are excited by light, transfer an electron per molecule to the encapsulated Fe(III) complex which is reduced in the process. The PAHs (orange) are regenerated by an external electron donor (EDTA). The reaction results in the net inward flow of electrons and an electron gradient is created.

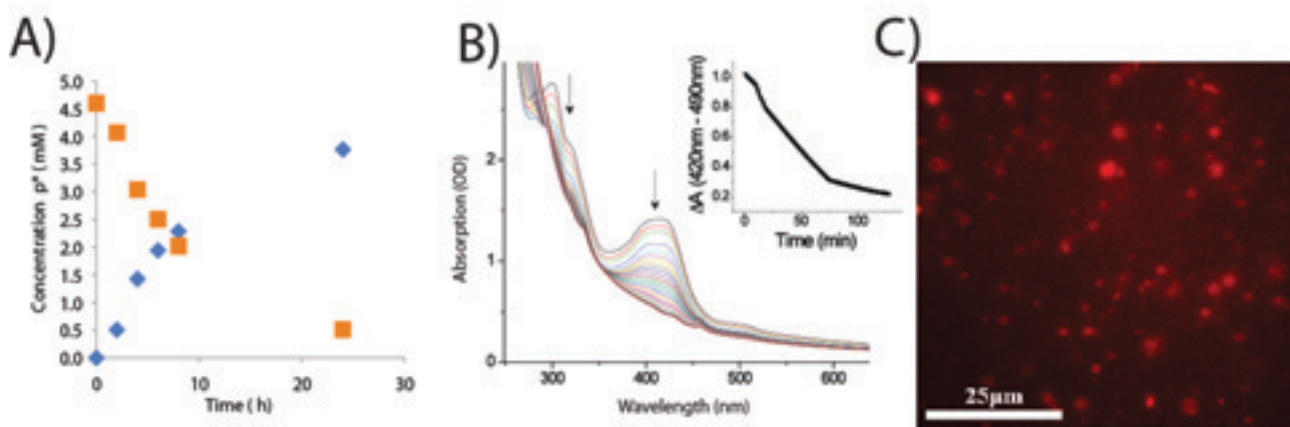


Figure 2: A) Photochemical production of amphiphiles. Blue diamond: concentration of free photocleavable group; Orange square: concentration of amphiphile precursor. B) Reduction of Fe III upon irradiation of the PAH. The arrow indicates the decrease of the Fe III specific absorption bands used to monitor reaction progresses. Insert: Wavelegth difference kinetics (420 nm minus 490 nm) of the reduction process. C) Micrograph of a fatty acid based protocellular population stained with Nile red.

In the latter case, we have designed fatty-acid/polycyclic aromatic hydrocarbon bilayer vesicles (Fig. 1b) that upon exposure to light are shuttling electrons from the external medium into the vesicular lumen (Cape *et al.*, 2011).

Both systems are implemented and functional as protocellular energy harvesters as can be seen in Fig. 2. In the ruthenium reaction, upon the removal of a photocleavable group fatty acid amphiphiles are produced and ligation reactions have been performed in bulk aqueous medium. The PAH supported electron transport across membranes was efficient.

We have previously reported theoretical and computational studies of free energy driven protocellular processes as well as full protocell life-cycles (Fellermann *et al.*, 2007, Rouchelau *et al.*, 2007, Munteanu *et al.*, 2007, Knutson *et al.*, 2008). In our current study, we focus on the interplay between the nonequilibrium metabolic driving and the equilibrium self-assembly processes and, in particular, the constraints these processes define for protocellular life-cycles and evolution. For example, we find that under a broad set of conditions, self-assembly processes necessary for the integrity of the protocellular container, equilibrate the selective advantage of a more efficient metabolic process in protocellular populations.

We report on our experimental and theoretical progress in understanding the intricacies of our protocellular model system's energetics.

References

- Cape, J., Monnard, P.-A., and Boncella, J. M. (2011) Prebiotically relevant mixed fatty acid vesicles support anionic solute encapsulation and photochemically catalyzed trans-membrane charge transport. *Chemical Sciences*, 2:661-667.
- Cape, J. L., Edson, J. B., Spencer, L. P., DeClue, M. S., Ziock, H.-J., Maurer, S. E., et al. Phototriggered DNA ligation using visible light in a tandem 5'-amine deprotection / 3'-phosphorimidazole coupling reaction. *Bioconjug Chem*, Submitted 2012.
- DeClue, M. S., Monnard, P.-A., Bailey, J. A., Maurer, S. E., Collis, G. E., Ziock, H.-J., et al. (2009) Nucleobase Mediated, Photocatalytic Vesicle Formation from an Ester Precursor. *J. Am. Chem. Soc.*, 131:931-933.
- Fellermann, H., Rasmussen, S., Ziock, H. & Solé, R. V. (2007) Life-cycle of a Minimal Protocell: a Dissipative Particle Dynamics (DPD) study. *Artificial Life*, 13(4):319-345.
- Knutson, C., Benko, G., Rocheleau, T., Mouffouk, F., Maselko, J., Chen, L., Shreve, A.P. & Rasmussen, S. (2008) Metabolic Photo-fragmentation Kinetics for a Minimal Protocell: Rate-limiting Factors, Efficiency and Implications for Evolution. *Artificial Life*, 14(2):189-201
- Maurer, S. E., DeClue, M. S., Albertsen, A. N., Dörr, M., Kuiper, D. S., Ziock, H., et al. (2011) Interactions between catalysts and amphiphile structures and their implications for a protocell model. *ChemPhysChem*, 12:828-835.
- Munteanu, A., Attolini, C., Rasmussen, S., Ziock, H. & Solé, R.V., Generic. (2007) Darwinian Selection in Catalytic Protocell Assemblies, *Philosophical Transactions of the Royal Society of London. Biological Sciences*, 362(1489):1847-1856.
- Rasmussen S., Bedau M., Chen L., Krakauer D., Packard N. and Stadler P. (2009) *Protocells*. MIT Press, Cambridge, MA.
- Rouchelau, T.; Rasmussen, S.; Nielsen, P.; Jacobi, M. & Ziock, H. (2007). Emergence of Protocellular Growth Laws. *Philosophical Transactions of the Royal Society of London. Biological Sciences*; 362(1486):1841-1845.
- Steinberg-Yfrach, G.; Rigaud, J.-L.; Durantini, E. N.; Moore, A. L.; Gust, D.; & Moore, T. A. (1998). Light-driven production of ATP catalysed by F₀F₁-ATP synthase in an artificial photosynthetic membrane. *Nature*, 392:479-482.

Programming DNA-based reaction-diffusion circuits for pattern transformation

Steven Chirieleison¹, Peter Allen¹, Andrew McIver², Alex Deiters², Andy Ellington¹, Xi Chen¹

¹Center for Systems and Synthetic Biology, University of Texas at Austin, Austin, TX 78712

²Department of Chemistry, North Carolina State University, Raleigh, NC 27695

xichen05@utexas.edu; andy.ellington@mail.utexas.edu

Extended Abstract

Generation and transformation of spatial patterns is a prevalent phenomenon in natural biological systems (Lander 2011) and an important goal in engineered biological systems (Basu et al. 2005). Other than genetic engineering, there is no system currently available to implement programmable reaction-diffusion kinetics and to design bottom-up systems capable of intricate pattern formation at the nanoscale and above. Here, we show the potential of DNA computation (Qian and Winfree 2011) for pattern generation and transformation by implementing amorphous computations (Nagpal 2001) in gel-based systems, culminating in the development of two classic patterning algorithms: an edge detector (Tabor et al. 2009) and an edge splitter (Fig. 1).

In theory, an incoherent feed forward loop (Fig. 2a) is necessary, and usually sufficient, to create an edge detector. To realize such an incoherent feed forward loop, we re-engineered a catalytic hairpin assembly (CHA) circuit (Yin et al. 2008; Li et al. 2011) (Fig. 2b, with a detailed molecular mechanism and experimental validation shown in 2c) so that exposure to UV activates catalysts (Fig. 3a) but inactivates substrates (Fig. 3b). Local to the incident light, the active catalyst has no substrate and therefore does not generate output at the site of activation. However, the catalyst can diffuse away from the site of activation, and at the edge of the light-dark boundary will encounter active substrates, initiating CHA and generating an edge.

We first confirmed the feasibility of photo-modulation of the catalyst and substrate. As expected, when the caged catalyst and a native substrate were used, a positive image was created (Fig. 4a). Similarly, when the native catalyst was used in conjunction with the inactivatable substrate, a negative image was observed (data not shown). When the caged catalyst and photocleavable hairpin substrate were used in the same reaction an incoherent feed forward loop was created, and the circuit generated an edge (Fig. 4b).

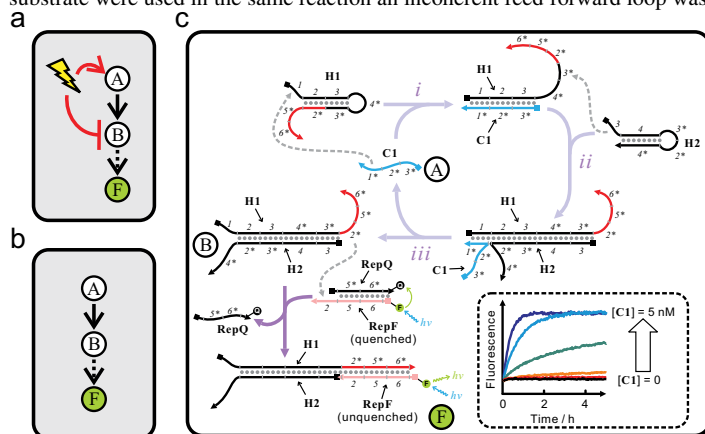


Figure 2. Schemes of the incoherent feed-forward loop (a) and the catalyzed hairpin assembly reaction (b and c).

pattern transformation circuits that mimic various phenomena in developmental biology such as the transformation of a chemical gradient into segments during insect embryogenesis. These circuits can potentially be used to program the positioning of molecules at macroscale in fields such as material science and tissue engineering

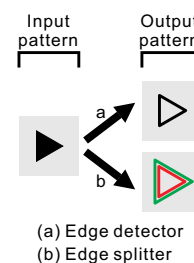


Figure 1.

To further explore the inherent programmability and modularity of our amorphous DNA computation, we sought to implement two orthogonal pattern transformation programs in the same gel. We first engineered a CHA circuit similar to that shown in Fig. 2c, but with a completely different set of sequences and a fluorescent reporter of a different color. We demonstrated that both the new positive image generator and the new edge detector could perform orthogonally, resulting in an overlaid pattern (Fig. 4c). Based on these successes, we next sought to engineer a more challenging pattern transformation program: an edge splitter. This program relied on programming the diffusivities of the species in the two orthogonal, edge-detection circuits. Different relative diffusivities should lead to different positions of the two edges, a split edge (Fig. 4d).

Finally, based on this eminently programmable reaction-diffusion network, we have designed other

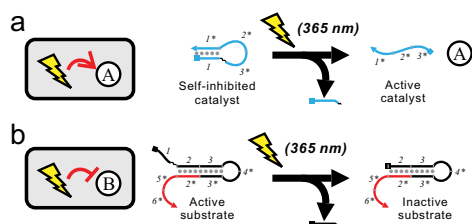


Figure 3. Activation of catalyst (a) and inactivation of substrate (b).

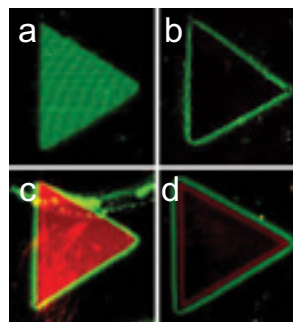


Figure 4. Patterns in a gel.

References

- Basu, S., Y. Gerchman, C. H. Collins, F. H. Arnold and R. Weiss (2005). A synthetic multicellular system for programmed pattern formation. *Nature*, 434:1130-1134.
- Lander, A. D. (2011). Pattern, growth, and control. *Cell*, 144:955-969.
- Li, B., A. D. Ellington and X. Chen (2011). Rational, modular adaptation of enzyme-free DNA circuits to multiple detection methods. *Nucleic Acids Res.*
- Nagpal, R. (2001). *Programmable Self-Assembly: Constructing Global Shape Using Biologically-Inspired Local Interactions and Origami Mathematics*. Ph.D. thesis, Department of Electrical Engineering and Computer Science, University
- Qian, L. and E. Winfree (2011). Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, 332:1196-1201.
- Tabor, J. J., H. M. Salis, Z. B. Simpson, A. A. Chevalier, A. Levskaya, E. M. Marcotte, C. A. Voigt and A. D. Ellington (2009). A synthetic genetic edge detection program. *Cell*, 137:1272-1281.
- Yin, P., H. M. Choi, C. R. Calvert and N. A. Pierce (2008). Programming biomolecular self-assembly pathways. *Nature*, 451:318-322.

A Generic Graphical Interface For Multicellular Simulation

Sylvain Cussat-Blanc, Jonathan Pascalie, Sylvain Tournois, Hervé Luga, Yves Duthen

Université de Toulouse - IRIT - CNRS - UMR 5505

sylvain.cussat-blanc@irit.fr, jonathan.pascalie@irit.fr, sylvain.tournois@gmail.com, herve.luga@irit.fr, yves.duthen@irit.fr

Extended Abstract

Introduction

Whereas cell-based models have very different simulation mechanisms, the graphical representation tends to be always the same. In order to save the work of graphical interface development, this paper presents a prototype of a generic multi-platform rendering tool for evo-devo models. This software has been developed to be usable with all kind of cell-based developmental model. It proposes a list of configurable cell states and animations put together in a simulation data file that describes the simulation story.

Description of the functioning

DevoCellPlayer is an open-source renderer for discrete time simulation. It needs to read simulation data aggregated in frames tagged by their simulation time top. Actually this software is only available for 2-D simulation, 3-D simulation being in most cases with continuous environment. The simulation space must be fixed for the entire simulation and it must be defined in 2-D. The number of cells can be unlimited.

This renderer is built to give offline-rendering of cell simulation. The purpose of building this kind of software is that it gives better means of interaction than a video. Videos are difficult to use as a developer help because they are expensive to generate (in computational terms) on the one hand and they do not allow step-by-step execution of the simulation on the other hand. Thus they are not suitable for a complete visual examination of the simulation results.

Owing to the specificities presented before on the one hand and on the Java implementation on the other hand, the allocated memory should be specified at the beginning of the simulation. The default allocated memory size seems to be enough for a large panel of 2-D multicellular simulation.

The next section presents the global functioning of DevoCellPlayer with its 3 needed files.

Split of the simulation into 3 files

This renderer is based on 3 human-readable files that describe the simulation. The *state file* describes all possible states of cells. It provides the properties (such as the name,

the color of the cell nucleus, of the cytoplasm and of the membrane) of all possible states. The colors are defined in the RGB system on 3 bytes. The background color of the simulation (i.e the environment color) is also defined in the state file. The *action file* describes all cell actions used in the simulation. Each action is linked to an animation that will provide its graphical representation. The current version proposes 15 of the most important animations (NOOP, mitosis, move, absorb, reject, apoptosis, etc.) of real cell cycle simulators. Figures 2 show the decomposition of *mitosis* and *absorption-rejection* animations. More animations can be easily added. Finally, the *simulation data file* presents the sequence of animations and states changes that happen during the simulation.

Functionalities

This software allows an *a-posteriori* visualization of an evo-devo simulation. This is an advantage because it is possible to navigate backward and forward on the simulation timeline. A direct access to every simulation step is then possible. Step-by-step progress in the timeline, simulation pause and stop are other possible controls. Zoom in and zoom out are also available to visualize more details on a subpart of the simulation. Chapters can be added in the simulation data

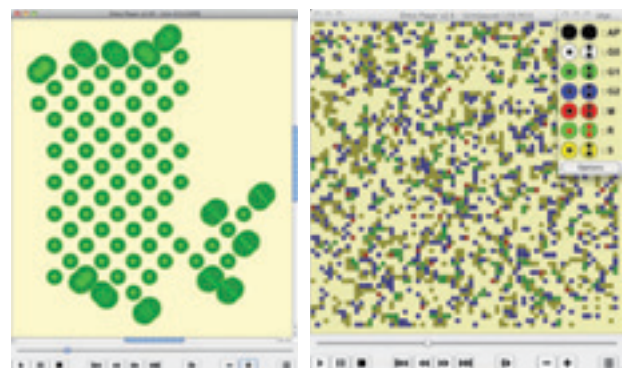


Figure 1: Examples of obtained visualization with the rendering tool and two different evo-devo models.

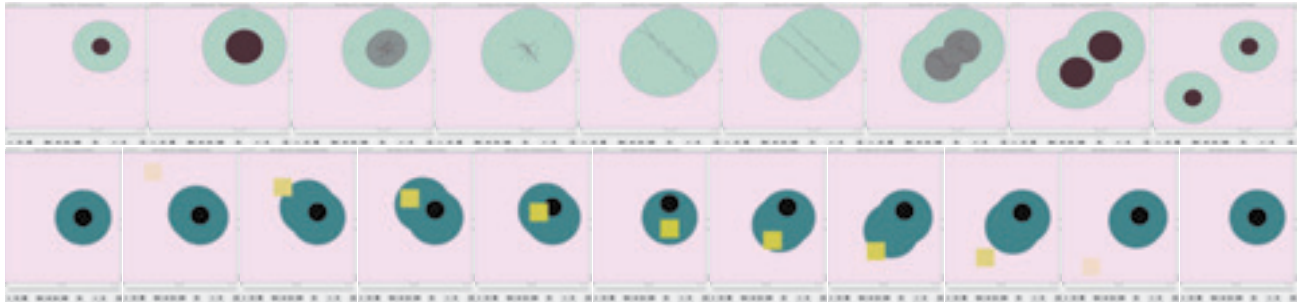


Figure 2: Line 1: The division animation: (a) Initial state; (b) mass doubling; (c-e) chromosome alignment; (f-h) mitosis; (i) final state. Line 2: The molecular exchange animation: (a-e) absorption of a molecule by a cell; (f) the molecule in the cell's cytoplasm; (g-k) rejection of the molecule.

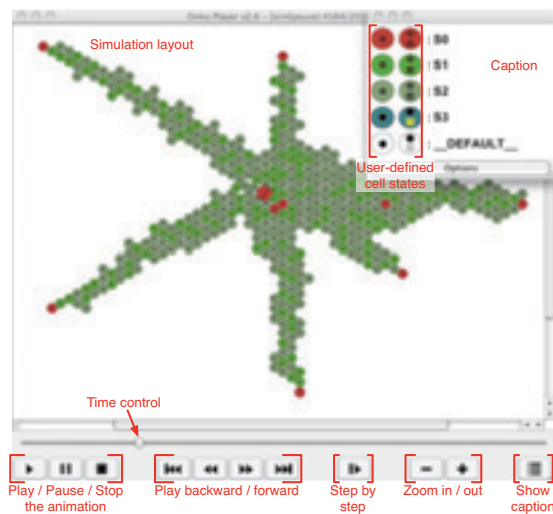


Figure 3: Details of the graphic interface of the player. It has been made as easy as possible to navigate as well in the timeline as in the space. The user can also define different states for the cells.

file in order to go directly to a particular event. Figure 3 shows a capture of the actual user interface.

The cell positioning is given thanks to absolute coordinates. It allows to define all kind of simulation grid as 8-neighbors 2-D grid or 6-neighbors 2-D grid. This positioning also allows to have a defined blank space between the cells or to represent heterogeneous environment with unusable site.

Cell's animations are also scheduled. At each time-step the advancement ratio of an animation is defined. This method allows to have cells with different executing speeds in their different animations. It also allows to stop an animation at every moment. In this way a cell arrested in mitosis can be represented.

Another functionality is to specify the cell shape. These

shapes are taken from a set of classic polygonal shapes like disk, square or triangle. The shape specification is embedded in the state file. Owing to the link between shapes and states, this renderer allow to have cells which are switching their shape during the simulation.

Conclusion and Future work

This prototype of simulation player has been tested on three very different evo-devo models developed in our research team. The integration was very easy and required less than one day of work. Figure 1 illustrates two examples of the use of the rendering tool with two of these models [1,2]. This software is actually available for downloading on the website: <http://www.irit.fr/devocellplayer>.

Lot of functionality can be added to this rendering software. The ergonomics of the renderer can be improved in order to get it closer to a conventional video player (simulation loading procedures, possible exports, etc.).

A multi-layer visualization could also be interesting to implement in order to visualize different aspects of the simulation (physical, chemical, decomposition of parallel tasks, etc.). Each layer has to be enough generic in order to represent all kind of simulations. Finally, this software will soon upgrade with a server mode. In this mode the frames will be sent in real time by the simulator to the software which will build the visualization. On the developer side, an interface could be imagined in order to connect directly the rendering tool to the simulator. The tool will keep all the features previously presented. The model could send each time step to the renderer, which builds on the fly the timeline. The step forward feature will not be accessible anymore but it will be possible to navigate in the past of the simulation.

References

- [1] S. Cussat-Blanc, H. Luga, Y. Duthen. From single cell to simple creature morphology and metabolism. In *Artificial Life XI*, pages 134–141. MIT Press, 2008.
- [2] J. Pascalie, V. Lobjois, H. Luga, B. Ducommun, Y. Duthen. A Checkpoint-Orientated Model to Simulate Unconstrained Proliferation of Cells. In *ECAL'11*, MIT Press, 2011.

Compartmentalized Partnered Replication (CPR): A Generalizable Method for the Evolution of Biomolecules

Jared W. Ellefson¹, Adam J. Meyer¹ and Andrew D. Ellington¹

¹Institute for Cellular and Molecular Biology, University of Texas at Austin
andy.ellington@mail.utexas.edu

Extended Abstract

There are many ways and many degrees by which artificial life can come into existence. Certainly artificial life-forms will carry traits which are not found in the sphere of natural life, as these characteristics can be more interesting than what we find in nature. Unnatural characteristics can be of great use, but we are left with the duty of finding these molecules in a vast and unforgiving sequence space. Tools to re-engineer or generate the stuff of life is, in many ways, the holy grail of artificial systems.

Compartmentalized Partnered Replication (CPR) is a generalizable method to evolve biomolecules that are linked to gene expression. The basis for the methodology was developed by the Holliger lab in the form of Compartmentalized Self Replication (CSR).¹ In CSR, a thermostable DNA polymerase could be evolved based its ability to replicate its own template through PCR. The new method, CPR, has been expanded to allow for the evolution of any functional sequence that can directly or indirectly lead to the expression of the DNA polymerase.

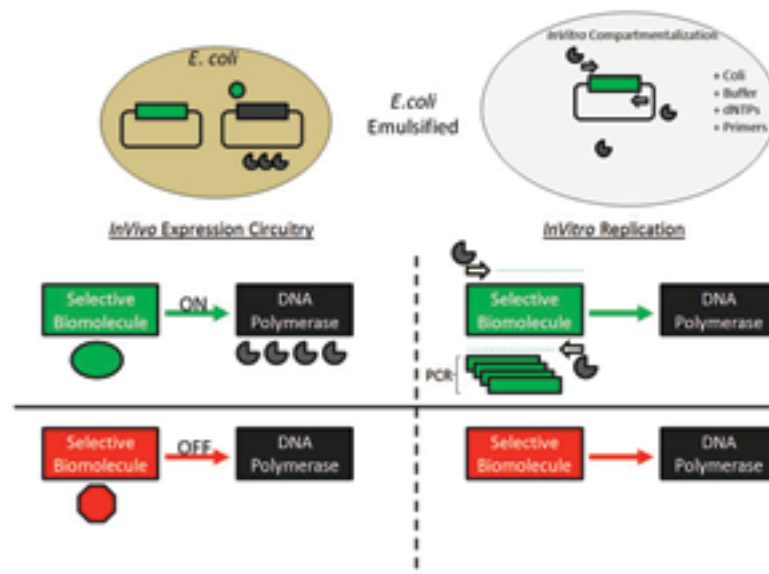
CPR is designed such that DNA polymerase expression is only achieved when the upstream circuitry displays the desired function. Cells harboring the circuit are emulsified with the components necessary for PCR, including primers that flank the circuit itself. Only cells harboring functional circuits imbue their compartments with the DNA polymerase necessary for PCR. Thus, upon thermal cycling functional circuits are amplified whilst inactive ones are not. A better adapted circuit results in higher DNA polymerase expression. This, in turn, results in greater amplification of the circuitry by PCR. Thus, the abundance of a given circuit in the population is directly proportional to the quality of the circuit. This direct linkage of phenotype and abundance allows for the evolution of a wide variety of biomolecules included DNA sequences, functional RNAs, and proteins.

For example, our method has been used to successfully enrich for functional T7 RNA polymerases, which in turn allow for the expression of the DNA polymerase. The CPR method has achieved a several hundred-fold enrichment for active T7 RNA polymerases after one round of selection. By altering the setup of the system, versions of T7 RNA polymerase have been evolved to recognize novel promoter sequences.

References

1. Ghadessy, F. J., Ong, J.L., and Holliger, P. (2001). Directed evolution of polymerase function by compartmentalized self-replication. *Proc Natl Acad Sci U S A.*, 98; 4552-4557.

Compartmentalized Partnered Replication (CPR)



Generation and screening of genomic libraries using *mariner* transposons and Cre/lox

Peter J. Enyeart¹, Jeffrey E. Barrick^{1,2}, Scott P. Hunicke-Smith³, Edward M. Marcotte^{1,2} and Andrew D. Ellington^{1,2}

¹Institute for Cell and Molecular Biology, ²Department of Chemistry and Biochemistry, and ³Genome Sequencing and Analysis Facility, University of Texas at Austin, Austin, Texas, 78712 USA
peter.enyart@utexas.edu

Extended Abstract

The overall structure of the bacterial genome is highly conserved, and it is generally considered that large rearrangements are unlikely to be tolerated (Rocha 2008). However, this view is based on studies involving a relatively small number of genomic inversions (Louarn et al. 1985; Hill and Gray 1988; Segall et al. 1988; Campo et al. 2004; Esnault et al. 2007), and the full search space of potential genomic rearrangements has not been extensively explored. We have devised a method for creating large libraries of bacterial strains with random genome rearrangements. These libraries are amenable to deep sequencing. We also apply the method toward deep-sequencing libraries of double mutants.

The methodology employed is shown in **Figure 1**. Transposons are employed to deliver resistance genes offset by *lox* sites to random locations in the genome. *Mariner* transposons are used due to their high efficiency and minimal site-insertion bias (Lampe et al. 1999; Rubin et al. 1999). The transposase is expressed from a plasmid while the transposons themselves are electroporated separately as non-replicating linear or circular DNA. We have achieved efficiencies of transposon integration of over $1 \times 10^5 \mu\text{g}^{-1}$, which allows creation of libraries having extensive coverage of the *E. coli* genome. Using transposons having different resistance elements allows selection for multiple insertions per genome. Once the transposons are inserted, expression of the Cre protein removes the markers and causes recombination between *lox* sites positioned in different regions of the genome. The removal of the resistance markers reduces the size of the scar to a size that allows the genomic sequence flanking both sides of the scar to be identified by deep sequencing.

We have created libraries of two and three transposon insertions per genome, and plan to have up to six transposons simultaneously present in a single genome. The libraries are subjected to selection pressure (e.g., serial growth in rich or minimal media), and deep sequencing is used to track the frequency of different library members over the course of the selection. If Cre is expressed prior to starting the experiment, a library of cells with rearranged genomes is subjected to selection. If Cre is expressed at the end of an experiment employing two transposons per genome, the genomic recombination allows the identities of the double mutants present to be determined. Our initial experiments focus on *E. coli* and compare a K strain (MG1655) and a B strain (REL606) in rich and minimal media. Given that both *mariner* transposons (Rubin, Akerley et al. 1999) and the Cre/lox system (Kilby et al. 1993) have been shown to function efficiently in a wide variety of organisms, including both prokaryotes and eukaryotes, the methodology presented herein should be widely applicable to many different biological systems.

References

- Campo, N., M. J. Dias, et al. (2004). Chromosomal constraints in Gram-positive bacteria revealed by artificial inversions. *Mol Microbiol* **51**(2): 511-522.
- Esnault, E., M. Valens, et al. (2007). Chromosome structuring limits genome plasticity in Escherichia coli. *PLoS Genet* **3**(12): e226.
- Hill, C. W. and J. A. Gray (1988). Effects of chromosomal inversion on cell fitness in Escherichia coli K-12. *Genetics* **119**(4): 771-778.
- Kilby, N. J., M. R. Snaith, et al. (1993). Site-specific recombinases: tools for genome engineering. *Trends Genet* **9**(12): 413-421.
- Lampe, D. J., B. J. Akerley, et al. (1999). Hyperactive transposase mutants of the Himar1 mariner transposon. *Proc Natl Acad Sci U S A* **96**(20): 11428-11433.
- Louarn, J. M., J. P. Bouche, et al. (1985). Characterization and properties of very large inversions of the E. coli chromosome along the origin-to-terminus axis. *Mol Gen Genet* **201**(3): 467-476.
- Rocha, E. P. (2008). The organization of the bacterial genome. *Annu Rev Genet* **42**: 211-233.
- Rubin, E. J., B. J. Akerley, et al. (1999). In vivo transposition of mariner-based elements in enteric bacteria and mycobacteria. *Proc Natl Acad Sci U S A* **96**(4): 1645-1650.
- Segall, A., M. J. Mahan, et al. (1988). Rearrangement of the bacterial chromosome: forbidden inversions. *Science* **241**(4871): 1314-1318.

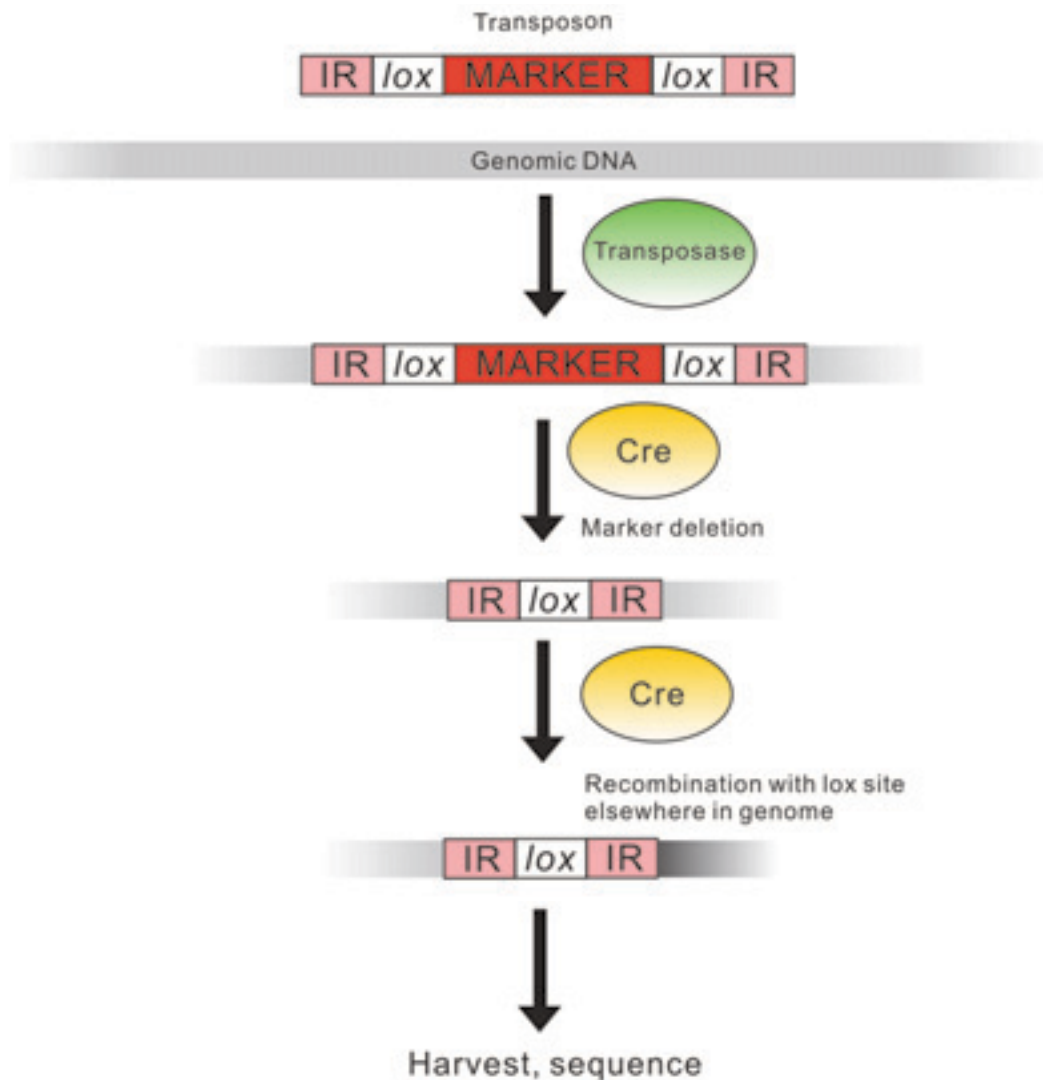


Figure 1. Methodology for delivering *lox* sites and screening genomic libraries. The "IR" box is the inverted repeat of the transposon; black is used to denote DNA originally from a different region of the genome than the grey genomic DNA.

An artificial multivesicular *in vitro* system to emulate multicellular processes

Maik Hadorn¹, Eva Boenzli¹, Martin M. Hanczyc¹, Steen Rasmussen^{1,2} and Peter Eggenberger Hotz³

¹Center for Fundamental Living Technology, Institute for Physics, Chemistry and Pharmacy, University of Southern Denmark, 5230 Odense M, Denmark

²Santa Fe Institute, Santa Fe NM 87501, USA

³Artificial Intelligence Laboratory, Department of Informatics, University of Zurich, 8050 Zurich, Switzerland
hadorn@sdu.dk

Extended Abstract

The three pillars of artificial life, i.e. simulation, hardware, and wetware, all exploit life's principles and aim for a more biologically inspired technology. Unlike the well-established liaison between hardware and simulations (Lichtensteiger and Eggenberger Hotz, 1999; Fraedrich and Goldberg, 2000; Hartland and Bredèche, 2006; Sargent, 2007; Zagal and Ruiz-del-Solar, 2007; Bacic et al., 2009), simulations that increasingly incorporate our current understanding of molecular and evolutionary biology (Banzhaf et al., 2006) are only marginally represented in current *in vitro* wetware systems. Eggenberger Hotz (2003) previously showed that simulated cells each containing a continuous genetic regulatory network are able to simulate the embryonic gastrulation process. Depending on the subset of active genes, a hollow sphere of tethered cells (Fig. 1a) is deformed by adjusting the strength of the adhesive viscoelastic elements (Fig. 1b). In order to emulate this gastrulation process with a novel multicompartment wetware system, artificial cells able to undergo cell division and differentiation are needed. However, the lack of such artificial living cells evokes the need for large sheets of surrogate artificial vesicles, that are highly organized in space and equipped with a basic cellular machinery.

Here, we present *in vitro* results on both the DNA-directed self-assembly of several types of artificial vesicles resulting in large sheets of assembled vesicles (Fig. 1c) as well as on the functionalization of their aqueous interior by a cellular machinery (Fig. 1d). The artificial vesicles were made from scratch with a phospholipid membrane functionalized with single stranded DNA oligonucleotides as adhesive element (for details of the protocol see (Hadorn and Eggenberger Hotz, 2010)). Furthermore, to prepare the aqueous compartments hosting either natural or synthetically designed genetic regulatory networks, we incorporated cell-free expression systems into the lumen of artificial vesicles to synthesize proteins *in vitro* (Fig. 1d) – a methodology described in the context of vesicle bioreactors by Noireaux and Libchaber (2004).

In our experiments, the positioning of vesicles in the large sheets of assembled vesicles was controlled only by local DNA interactions. Consequently, the highly organized architecture proposed in Figure 1e has not been achieved so far. However, Figures 1e to 1h detail a concept for the further implementation of a wetware system able to emulate the simulated gastrulation process. Within a two-layered sheet of artificial vesicles, a distinct population of vesicles, localized at the center of the sheet (Fig. 1e, green), bears both the genetic blueprint and the cellular machinery able to synthesize a digestive enzyme (e.g. amylase). Furthermore, a polymeric substrate of the enzyme is incorporated in the aqueous interior (e.g. starch). After the digestive enzyme is synthesized (Fig. 1f), the substrate is broken down into a large number of smaller units (e.g. glucose, Fig. 1h), which increase the osmotic pressure inside, induce an influx of water, and consequently increase the volume of the vesicles. The increase of the volumes of some parts of the sheet induces mechanical stress forcing a bending of the two-dimensional sheets (Fig. 1g), which emulates both the simulated and the natural gastrulation process.

One may argue that emulating natural processes – like the embryonic gastrulation – without employing the actual cellular components only shallowly portrays the processes in the natural model. However, because our system is designed to exploit inherent material properties of the components it offers insights how nature exploits properties that are implicitly stored in the genetic blueprint. Additionally, our system offers an increase in complexity on demand by incorporating more and more functionalities that characterize natural cells (e.g. cell division, cell differentiation). The proposed system may be adjusted over time to emulate additional features of natural organisms. This well-controlled, self-assembled, minimal artificial cell system exploits implicit material properties, and when equipped with appropriate cell-free expression systems it may be a valid basis for new experimental tools to test critical mechanisms involved in more complex organismic biological processes.

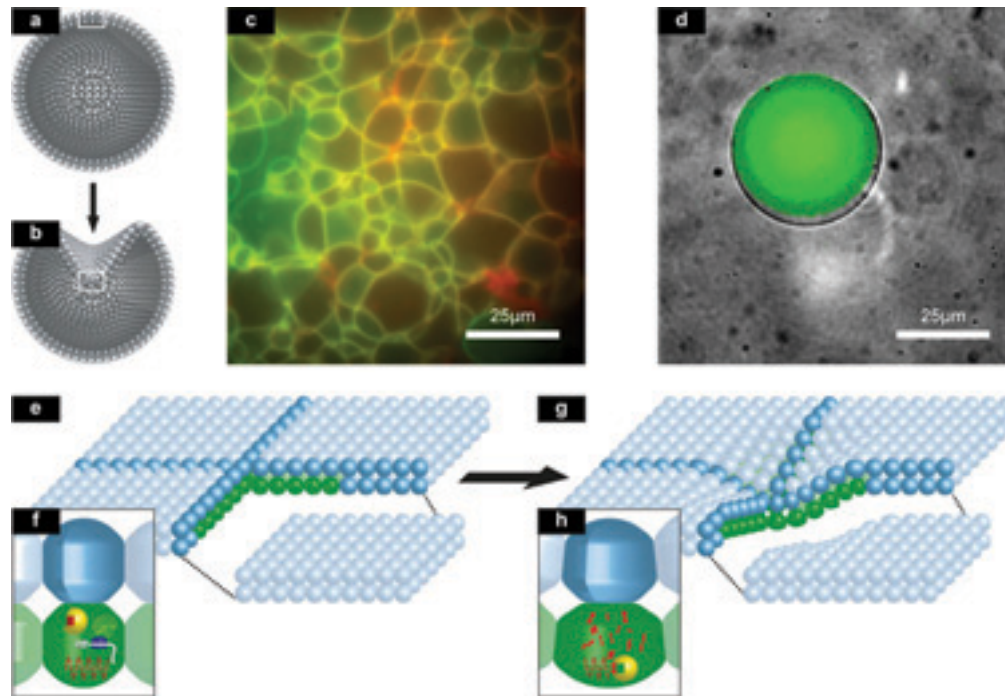


Figure 1: (a,b) Sketch of the simulated gastrulation process. (c,d) Fluorescence and transmission micrographs of the wetware experimental results on large sheets of artificial vesicles (c) and on gene expression in the interior of artificial vesicles; here visualized by the expression of green fluorescent protein (d). (e-h) Proposed outline of the implementation of a wetware system able to emulate both the simulated and the natural gastrulation process. See text for details.

References

- Bacic, M., S. Neild and P. Gawthrop (2009). Introduction to the special issue on hardware-in-the-loop simulation. *Mechatronics*, 19: 1041-1042.
- Banzhaf, W., G. Beslon, S. Christensen, J. A. Foster, F. Kepes, V. Lefort, J. F. Miller, M. Radman and J. J. Ramsden (2006). Guidelines - From artificial evolution to computational evolution: a research agenda. *Nature Reviews Genetics*, 7: 729-735.
- Eggenberger Hotz, P. (2003). Combining developmental processes and their physics in an artificial evolutionary system to evolve shapes. In *On Growth, Form and Computers*. S. Kumar and P. Bentley, editors. Academic Press: 302-318.
- Eggenberger Hotz, P. editors (2004). *Asymmetric cell division and its integration with other developmental processes for artificial evolutionary systems*. MIT Press, Cambridge.
- Fraedrich, D. and A. Goldberg (2000). A methodological framework for the validation of predictive simulations. *European Journal of Operational Research*, 124: 55-62.
- Hadorn, M. and P. Eggenberger Hotz (2010) DNA-Mediated Self-Assembly of Artificial Vesicles. *Plos One* 5, e9886 DOI: 10.1371/journal.pone.0009886.
- Hartland, C. and N. Bredèche editors (2006). *Evolutionary robotics, anticipation and the reality gap*. IEEE, New York.
- Lichtensteiger, L. and P. Eggenberger Hotz (1999). Evolving the Morphology of a Compound Eye on a Robot. In editors *Proceedings of the Third European Workshop on Advanced Mobile Robots (Eurobot '99)*, 127-134. IEEE.
- Noireaux, V. and A. Libchaber (2004). A vesicle bioreactor as a step toward an artificial cell assembly. *Proceedings of the National Academy of Sciences of the United States of America*, 101: 17669-17674.
- Sargent, R. G. editors (2007). *Verification and validation of simulation models*. IEEE, New York.
- Zagal, J. C. and J. Ruiz-del-Solar (2007). Combining simulation and reality in evolutionary robotics. *Journal of Intelligent & Robotic Systems*, 50: 19-39.

A Minimal Artificial Subcellular Matrix

Maik Hadorn¹, Benny Gil⁵, Carsten Svaneborg¹, Martin M Hanczyc¹, Harold Fellermann¹, Rudolf Fuchslin⁴, Peter Eggenberger Hotz³, Casper Kunstmann-Olsen¹, Doron Lancet⁶, John McCaskill⁸, Pierre-Alain Monnard¹, Gunter von Kiedrowski⁷ & Steen Rasmussen^{1,2}

¹Center for Fundamental Living Technology (FLinT), University of Southern Denmark, Denmark

²Santa Fe Institute, Santa Fe NM, USA

³Artificial Intelligence Laboratory, University of Zurich, Switzerland

⁴European Center for Living Technology, Venice, Italy

⁵Computer Science and Applied Mathematics & ⁶Molecular Genetics, Weizmann Institute of Science, Rehovot, Israel

⁷Bioorganic Chemistry & ⁸BioMIP, Ruhr University Bochum, Germany

steen@sdu.dk

Extended Abstract

A biological subcellular matrix functions through an intricately coordinated material transportation, information processing and material production system. We seek to mimic these fundamental properties utilizing a hybrid biochemical and information technological system. We introduce an integrated programmable information- and production chemistry by having DNA addressable chemical containers (chemtainers) interfacing traditional electronic computers via microelectromechanical systems (MEMS) with regulatory feedback loops¹ (Amos et al., 2011). DNA tags anchored in the chemtainers make them addressable with respect to each other through complementary DNA interaction as well as addressable within a MEMS microfluidics matrix through DNA tags anchored in the micro fluidics channels.

A representative snapshot of the different types of chemtainers employed is shown in figure 1 with DNA nano-cages, vesicles (lipid and fatty acid), oil-in-water emulsion droplets and water droplets in ionic liquids. The micro fluidic MEMS matrix with immobilized single stranded DNA (schematic in figure 1) represents the interface between the chemtainers and the electronic computers by controlling the attachment of DNA-coated chemtainers.

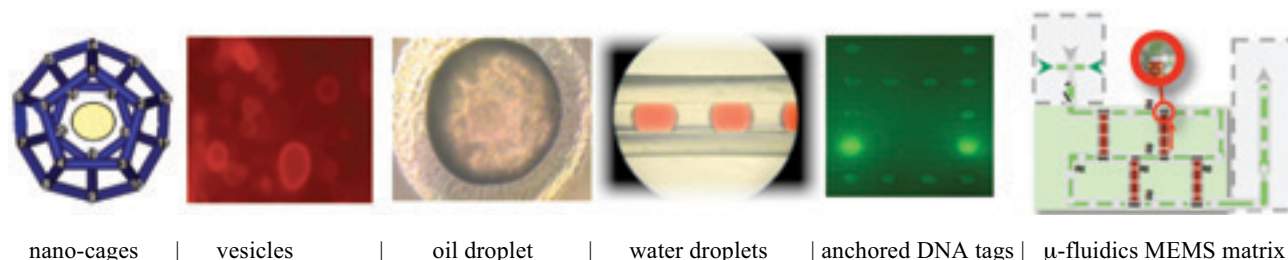


Figure 1. A representative sampling of addressable chemical containers (chemtainers) in a micro fluidics MEMS environment interfacing digital computers with a sensor input to micro fluidics electronic feedback.

The abovementioned chemtainers vary significantly in terms of scale and functionality. At the nanoscale, DNA single strands are both the building blocks of the containers and the instance to functionalize them. These DNA cages can open and close controlled by external signals and when closing encapsulate macromolecules as cargo. At the microscale, the DNA is not used as building material but to address the surface of the chemtainers. These microscopic chemtainers act as either hydrophilic or hydrophobic reaction vessels, which can themselves determine their next processing steps. DNA labeling and addressing of the larger water droplets is also possible (Wagler et al., 2012). DNA-directed fusion of chemtainers will replace fusion events already shown to be triggered by electrostatic interactions between artificial vesicles (Caschera et al., 2011).

A key point for all these technologies is the use of DNA addresses to coordinate the specific assembly of chemtainers in space and time. As an example, we have developed a modular DNA addressing system for supramolecular chemtainers. DNA single strands are incorporated into the surface both of artificial vesicles (Hadorn and Eggenberger Hotz, 2010) and of oil-in-water emulsion droplets (Hadorn et al., 2012). In this way we can program the assembly of chemtainers using local base pairing rules, see Figure 2a for a representative micrograph of assembled oil-in-water emulsion droplets. Both the sequence and length of the DNA addresses can be modified to ensure both specificity and robust hybridization against denaturing thermal effects (Chan et al., 2007). The same

¹ Matrix for Chemical IT (MATCHIT), see <http://www.fp7-matchit.eu>

methodology directing the assembly of chemtainers is applied to immobilize them to a solid support. Conditions that disfavor the DNA base pairing (i.e. increase of temperature, decrease in salt concentration, addition of competitive DNA) is used to reverse the assembly process of chemtainers. In addition, we have demonstrated that the DNA addresses can be detached from the surface and replaced by new addresses. This allows for altered programmed assembly and a recyclability of our system. Dissipative Particle Dynamics (DPD) simulations of oil-droplets tagged with DNA molecules, using a novel dynamic bonding DNA model (Svaneborg, 2012), is shown in Figure 2(b), where complementary addressed chemtainers associate specifically and then fuse.

Using DNA addresses a common language of the diverse types of chemtainers combined with chemical reactions controlled by programmable fusion of chemtainers opens up for a new kind of computing. This computing allows parallel chemical and internal material production programming in a multilevel architecture. Through autonomous DNA address modification (utilizing the usual DNA computing operation) and resolution at the container-container, container-surface, and container-molecule levels, the architecture provides a concrete embedded application for integrated information processing, computing and material production. Self-organizing container addressing will allow micro- and nanoscale processing of any collection of chemicals that can be packaged in the containers. We are developing a calculus that expands but closely follows the line of brane calculi for expressing nested membrane systems (Cardelli, 2004). The extension to the brane calculus is necessary to accommodate the electronic feedback between the chemtainers and the monitoring-actuating MEMS matrix as well as the spatial addressing. The calculus can both be used for modeling chemtainers, chemtainer addressing and -interactions, as well as ultimately programming the microfluidic device. Elements of the calculus are (possibly nested) chemtainer systems, their cargo, and address tags. Operations of the calculus include chemtainer attachment, fusion, and cargo separation.

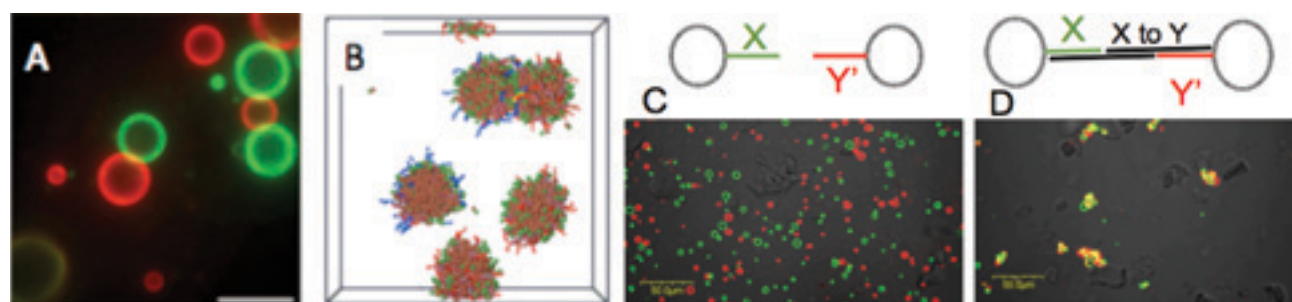


Figure 2. Implementation of Chemical IT. (A) Supramolecular oil-in-water emulsion droplets assembled by local DNA base pairing rules (red to green). (B) DPD simulation also supports assembly of droplets by local rules (blue to red). (C) Two populations of DNA tagged vesicles do not interact because of lack of DNA base pairing. (D) DNA computing used for address modification cause the vesicle populations to assembly due to DNA base pairing.

By exploiting the latest advances in electronic and biomaterial systems, we aim to create a hybrid machine/chemistry system for next generation artificial life technologies, ChemBio-ICT. Our approach constitutes a hybrid bottom up construction of life-like and living systems.

References

- Amos M, Dittrich P, McCaskill J and Rasmussen S. (2011). Biological and Chemical Information Technologies, *Procedia Computer Science* 7, 56-60.
- Cardelli L. (2004). Brane Calculi. In Vincent Danos and Vincent Schächter (Eds.) CMSB'04: Proceedings of the 2nd international workshop on Computational Methods in Systems Biology, volume 3082 of Lecture Notes in Bioinformatics, pages 172-191. Springer-Verlag.
- Caschera, F., T. Sunami, T. Matsuura, H. Suzuki, M. M. Hanczyc and T. Yomo (2011). Programmed Vesicle Fusion Triggers Gene Expression. *Langmuir*, 27: 13082-13090.
- Chan Y-H.M., Lenz P., and Boxer S.G. (2007) Kinetics of DNA-mediated Docking Reactions Between Vesicles Tethered To Supported Lipid Bilayers", *PNAS*, 104, 48, 18913-18918.
- Gill, B; Kahan-Hanum M; Skirtenko, N; Adar R. and Shapiro E. Docter in a cell: Vision and Accomplishments: In Proceedings of the Artificial Life 12 Conference (2010)
- Hadorn M, Eggenberger Hotz P. (2010). DNA-Mediated Self-Assembly of Artificial Vesicles. / *PLoS One* 5 5(3):e9886.
- Hadorn M., Bonzli E., Eggenberger Hotz P., and Hanczyc M.M. (2012). Programmable and Reversible DNA-directed Self-Assembly of Emulsion Droplets. *In review*.
- Svaneborg C. (2012) LAMMPS framework for Dynamic Bonding and an Application Modelling DNA. *Comp. Phys. Comm. Accepted*.
- Wagler, P.F., Tangen, U., Maeke, T. and McCaskill, J.S. Field programmable chemistry: Integrated chemical and electronic processing of informational molecules towards electronic chemical cells, *Biosystems*, in press, 2012.

Towards protocell embedded replication of nucleic acids

Philipp M.G. Löffler¹, Rafal Wieczorek¹, Michael Wamberg¹, Mark Dörr¹, Pernille L. Pedersen¹, Carsten Svaneborg¹, Harold Fellermann¹, Joseph B. Edson², Jonathan L. Cape², Hans Ziock³, James M. Boncella², Steen Rasmussen^{1,4}, Pierre-Alain Monnard¹

¹Center for Fundamental Living Technology (FLinT), Department of Physics, Chemistry & Pharmacy, University of Southern Denmark, 5230 Odense M, Denmark; ²Materials, Physics and Applications & ³Earth and Environmental Science, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA; ⁴Santa Fe Institute, Santa Fe NM 87501 USA
monnard@sdu.dk

Extended Abstract

The construction of a self-reproducing chemical system possessing simplified capabilities of a living cell is a major scientific challenge and a milestone for Artificial Life research. Minimal living physicochemical systems should encompass “information”, “energy transformation” and “co-localization” in a mutually interdependent manner. A fundamental design strategy to achieve such minimal living system was outlined in Rasmussen et al. (2003, 2004), and to reach that goal we have defined a chemical dependency between each of the three component subsystems. The replication of the information molecules must depend on the formation of the container, while the formation of the container must depend on the work of the metabolism, likewise, the work of the metabolism must depend on the replication of the information molecules. This work focuses on the information replication processes, where we have recently developed a template-directed synthesis of oligonucleotides by non-enzymatic ligation (Cape et al., 2012), which is connected to the growth of the container by the action of a Ruthenium catalyst (Fig. 1). In earlier studies we have demonstrated how the presence of a particular nucleobase (8-oxoguanine) - out of a possible combinatorial set - controls the growth metabolism (DeClue et al., 2009 & Maurer et al., 2011).

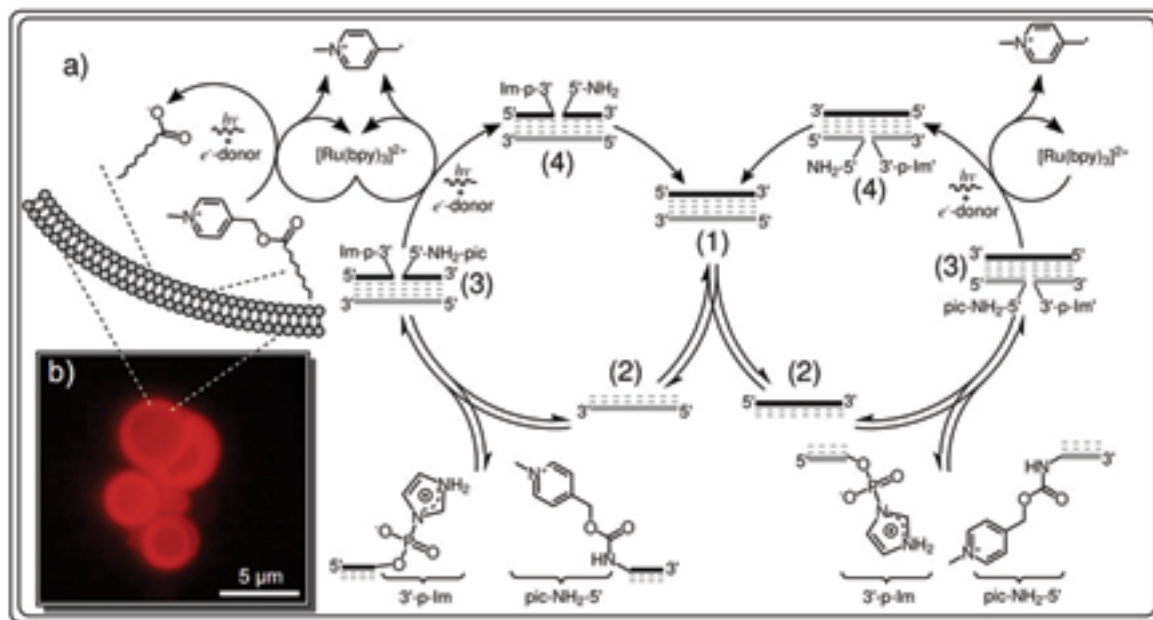


Figure 1. (a) Schematic of non-enzymatic strand replication mechanism (information) coupled with amphiphile production (container) via the light driven Ruthenium catalysis (metabolism). The nucleic-acid cross replication evolving from the double stranded template (1) proceeds by dehybridization (1-2), hybridization of oligomer building blocks, (2-3), light-triggered deprotection of the 5'-amino group (3-4) and ligation (4-1). The electron donor is either 8-oxoguanine or ascorbic acid. (b) Micrograph of vesicles formed during simultaneous light-driven conversion of the amphiphile precursor and ligation of DNA oligomers, both catalyzed by the Ruthenium complex.

As illustrated on Figure 1a, our starting point is the double-stranded template that guide the hybridization of shorter oligonucleotides (1-3), one having an activated 3'-phosphate and one having the protected 5'-amino group (Edson et al. 2011). Only after the light-triggered deprotection catalyzed by the Ruthenium complex under the action of an electron donor (8-oxoguanine or ascorbic acid) (3-4) the ligation of the shorter oligonucleotides can occur. Thus, the double-stranded DNA

product (1) is formed, closing the reaction cycle (see Figure 1). The proof of principle was shown by a light-triggered deprotection and subsequent ligation of a 13-mer 5'-amino-oligomer in the presence of a self-priming hairpin (55 nt) with a 3'-imidazolylphosphate, monitored by High-Pressure Liquid Chromatography (HPLC) (Cape et al., 2012). At the same time, Ruthenium also catalyzes a similar reaction that converts an amphiphile precursor into new surfactant, which drives container growth. Both reactions were performed concurrently within one protocell population, demonstrating that a single metabolic step can produce container building blocks and novel copies of the information molecules. Figure 1b shows a micrograph of the reaction mixture after 3h irradiation with light (400 nm, 20nm band monochromator). During the reaction, Dynamic Light Scattering (DLS) measurements confirmed the formation of more vesicular structures.

Good nucleic acid yields for non-enzymatic replication are difficult to obtain and product inhibition is known to be one of the main causes of this problem. We therefore conduct theoretical and simulation studies of possible replication strategies to optimize the replication rate. One strategy is to cycle the temperature around the T_m (melting) of the template-product complex with an excess of the shorter oligomers. Figure 2a shows a simulations of non-enzymatic template based replication reactions using a novel 3D Langevin dynamic bonding framework (Svaneborg, 2012), where we can study sequence specific melting and renaturing transitions as well as template based ligation reactions. In particular, we apply these techniques to study the effects of base-pair binding strength, strand length, bulk vs. surface bound templates and different temperatures as well as temperature cycles. Earlier studies (Fellermann & Rasmussen, 2011) for varied temperatures and strand lengths (Fig. 2b) indicated a clear replication advantage for longer strands at lower temperatures in the regime where the ligation rate is rate limiting. In addition, these results indicate the existence of an optimal replication rate at the boundary between the two regimes where the ligation rate and the dehybridization rates are rate limiting (long strands and low temperatures). The efficiency of these replication schemes is currently being tested experimentally.

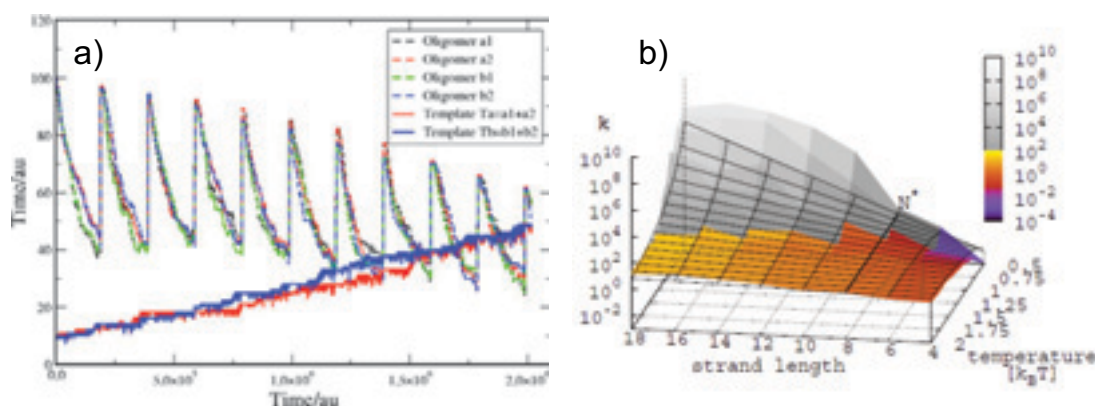


Figure 2. (a) The time evolution of molecular species in a system comprising two complementary templates and four oligomers. Periodically the temperature is raised to melt hybridized templates and oligomers. Progressively the oligomers are consumed to produce new templates. (b) Resulting replication rate k as a function of template length and temperature (without temperature cycling) under the assumption that ligation is rate limiting (slowest reaction). Note that k increases with template length and lower temperatures.

References

- Cape, J.L., Edson, J.B., Spencer, L.P., DeClue, M. S., Ziock, H., Maurer, S.E., Rasmussen, S., Monnard, P.A. and Boncella, J.M. (submitted). Phototriggered DNA ligation using visible light in a tandem 5'-amine deprotection / 3'-phosphorimidazolide coupling reaction.
- DeClue, M. S., Monnard, P. A., Bailey, J. A., Maurer, S. E., Collis, G. E., Ziock, H. J., Rasmussen, S. and Boncella, J.M. (2009). Nucleobase Mediated, Photocatalytic Vesicle Formation from an Ester Precursor. *J. Am. Chem. Soc.*, 131:931-933.
- Edson, J.B., Spencer, L.P., and Boncella, J.M. (2011). Photorelease of Primary Aliphatic and Aromatic Amines by Visible-Light-Induced Electron Transfer. *Org. Letters*, 13:6156-6159.
- Fellermann, H., Rasmussen, S. (2011). On the Growth Rate of Non-Enzymatic Molecular Replicators. *Entropy*, 13:1882-1903
- Maurer, S.E., DeClue, M.S., Albertsen A.N., Dorr, M., Kuiper, D.S., Ziock, H., Rasmussen, S., Boncella, J.M. and Monnard, P.A. (2011). Interactions between catalyst and amphiphilic structures and their implications for a protocell model. *Chemphyschem.*, 12:828-35.
- Rasmussen, S., Chen, L., Nilsson, M. and Abe, S. (2003). Bridging nonliving and living matter. *Artif Life*, 9:269-316.
- Rasmussen, S., Chen, L., Deamer, D.W., Krakauer, D.C., Packard, N.H., Stadler, P.F. and Bedau, M.A. (2004). Transitions from nonliving to living matter. *Science*, 303:963-965
- Svaneborg, C. (in press). LAMMPS framework for Dynamic Bonding and an Application Modelling DNA. To appear in the *Comp. Phys. Comm.*

Creating an artificial cell with different size revealed the effect of compartment volume on the intracompartamental multimeric protein synthesis

Tomoaki Matsuura^{1,3}, Kazufumi Hosoda², Norikazu Ichihashi³, Hiroaki Suzuki^{2,3} and Tetsuya Yomo^{2,3}

¹Department of Biotechnology, Osaka University, Yamadaoka 2-1, Suita, Osaka, Japan

²Department of Bioinformatic Engineering, Osaka University, Yamadaoka 1-5, Suita, Osaka, Japan

³ Exploratory Research for Advanced Technology, Japan Science and Technology Agency, Yamadaoka 1-5, Suita, Osaka, Japan

matsuura_tomoaki@bio.eng.osaka-u.ac.jp

Extended Abstract

Attempts have been made to construct artificial systems that mimic the biological ones only from defined components (Ichihashi, et al., 2010). This approach, so called the bottom-up approach (Jewett & Forster, 2010; Simpson, 2006), is expected to elucidate the properties of biological system, which is difficult to investigate with living cells. In the current study, using such bottom-up approach, we aimed to investigate the effect of cell volume on the intracellular reaction.

Cells change their size and shape depending on the cell cycle and the external environment (Lang, et al., 1998; Lizana, et al., 2009), however, it remains unclear how the cell volume alone affects the intracellular reaction. The effect of volume could be studied systematically and quantitatively if we could actively alter a cell's size and investigate the effect on intracellular reactions (Lizana, et al., 2009). Strategies to alter cell size include the application of osmotic pressure (Klipp, et al., 2005), but osmotic swelling dramatically changes the internal state of the cell (e.g., salt concentrations), so this strategy is not ideal for investigating only the effects of cell volume on the intracellular reactions. To study the fundamental response of the intracellular reaction to the cell volume, it would be useful to design a compartment that can alter its constituents and size as desired (Lizana, et al., 2008).

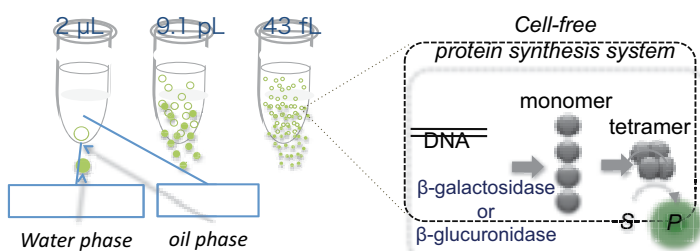


Figure: Schematic of multimeric protein synthesis using cell-free protein synthesis system in microcompartment with different volume. GAL and GUS syntheses was done inside the water-in-oil (w/o) emulsion droplets with an average volume of 2 μ L, 9.1 pL, and 43 fL. The syntheses within the emulsion are detected as an increase in the fluorescence signal that is obtained through a reaction cascade consisting of transcription, translation, monomer-to-tetramer assembly, and fluorescence substrate hydrolysis.

In this study, we used water-in-oil emulsion as a microcompartment whose droplet volume can be varied as desired, and encapsulated a cell-free protein synthesis system (Matsuura, et al., 2011), gene encoding β -glucuronidase (GUS) or β -galactosidase (GAL), both of which are homo-tetramer, and fluorescence substrate of respective enzyme inside the emulsion (Figure). The synthesis of tetrameric GUS and GAL was followed as an increase in the fluorescence signal in emulsion with different size ranging from 27 fL to 2 μ L. We found that production of GUS become faster as the size of the compartment decrease. Such acceleration was not observed with GAL. We found that the difference between GUS and GAL synthesis is caused by the difference in their rate-limiting step. Tetrameric GUS is produced faster in smaller compartments because the rate-limiting step is monomer-to-tetramer assembly, whereas tetrameric GAL is produced equally in both large and small compartments, because the assembly is so fast that monomers assemble into tetramers as soon as they are synthesized. As most of the biochemical reaction involves the complex formation, such principle applies to almost all components and reactions inside the cell. Our results also suggest that smaller cells may be beneficial for producing the functional forms of multimeric proteins for which the rate-limiting step is multimerization. Efficient production of multimers in smaller compartments might have played an essential role on the evolution of primitive cells (Pohorille & Deamer, 2002; Rasmussen, et al., 2004). Furthermore, our results suggest the importance of taking into account the compartment size for the construction of artificial cells (Ichihashi, et al., 2010; Szostak, et al., 2001).

References

- Ichihashi N, Matsuura T, Kita H, Sunami T, Suzuki H, Yomo T (2010) Constructing partial models of cells. *Cold Spring Harb Perspect Biol* **2**: a004945.
- Jewett MC, Forster AC (2010) Update on designing and building minimal cells. *Curr Opin Biotechnol* **21**: 697-703.
- Klipp E, Nordlander B, Kruger R, Gennemark P, Hohmann S (2005) Integrative model of the response of yeast to osmotic shock. *Nat Biotechnol* **23**: 975-982.
- Lang F, Busch GL, Ritter M, Volkl H, Waldegger S, Gulbins E, Haussinger D (1998) Functional significance of cell volume regulatory mechanisms. *Physiol Rev* **78**: 247-306
- Lizana L, Bauer B, Orwar O (2008) Controlling the rates of biochemical reactions and signaling networks by shape and volume changes. *Proc Natl Acad Sci U S A* **105**: 4099-4104.
- Lizana L, Konkoli Z, Bauer B, Jesorka A, Orwar O (2009) Controlling chemistry by geometry in nanoscale systems. *Annu Rev Phys Chem* **60**: 449-468.
- Matsuura T, Hosoda K, Ichihashi N, Kazuta Y, Yomo T (2011) Kinetic analysis of beta-galactosidase and beta-glucuronidase tetramerization coupled with protein translation. *J Biol Chem* **286**: 22028-22034.
- Pohorille A, Deamer D (2002) Artificial cells: prospects for biotechnology. *Trends Biotechnol* **20**: 123-128.
- Rasmussen S, Chen L, Deamer D, Krakauer DC, Packard NH, Stadler PF, Bedau MA (2004) Evolution. Transitions from nonliving to living matter. *Science* **303**: 963-965.
- Simpson ML (2006) Cell-free synthetic biology: a bottom-up approach to discovery by design. *Mol Syst Biol* **2**: 69.
- Szostak JW, Bartel DP, Luisi PL (2001) Synthesizing life. *Nature* **409**: 387-390.

An evolutionary-genomics approach for elucidating and improving complex microbial phenotypes

Jeremy Minty¹, Jihyang Park¹, Harris Wang², Lawrence Lai¹, Ted Zaroff III¹, Brian Johnson¹, Mark Burns¹, George Church², and Xiaoxia Nina Lin¹

¹Department of Chemical Engineering, University of Michigan

²Department of Genetics, Harvard Medical School

ninalin@umich.edu

Extended Abstract

Understanding the genetic architecture of complex phenotypes is of great fundamental interest and has important ramifications in medicine and biotechnology. Metabolic engineering efforts have enabled microbial production of many fuels and commodity chemicals, but frequently toxicity limits production. Microbial stress tolerance is a complex multigenic trait intractable to traditional genetic study and rational engineering efforts. Most approaches to improving stress tolerance are therefore combinatorial, following a strategy of generating diversity in a population and characterizing isolates with desired properties. However, present methods explore relatively small genotype spaces and often fail to capture epistatic interactions between distal genetic loci. We are developing an evolutionary-genomics methodology that transcends many limitations in previous approaches. The essence of our approach entails experimental evolution of stress tolerance followed by genome re-sequencing to identify acquired mutations, genomic and functional dissection to reverse engineer mechanisms of tolerance, and targeted genome engineering and high throughput screening for further phenotype improvement (Figure 1).

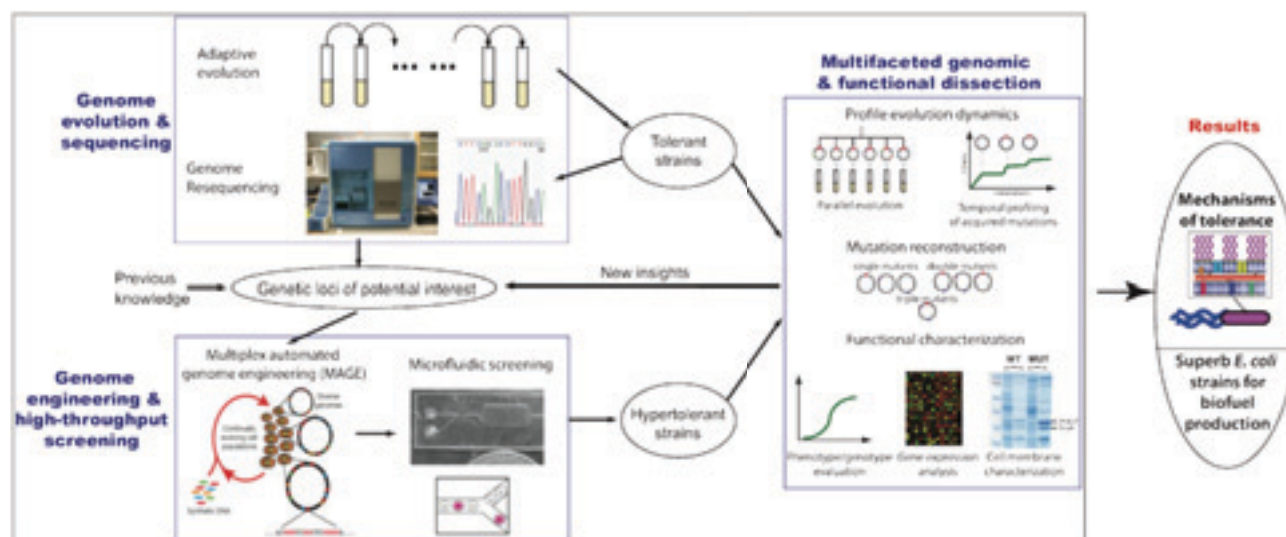


Figure 1: Overview of an evolutionary-genomics approach for tolerance phenotype elucidation and engineering.

As proof-of-concept, we applied this approach to investigate and improve *E. coli* tolerance to isobutanol, a promising next-generation biofuel (Minty, et al. 2011). We experimentally evolved multiple *E. coli* lineages on isobutanol spiked media for approximately 500 generations, resulting in up to 60% improvement in the minimum inhibitory concentration (MIC). We performed genome re-sequencing on highly tolerant isolates, followed by subsequent investigations including surveying parallel evolution and temporal genotypic dynamics across different populations, reconstructing key mutations in the parent *E. coli* strain to study phenotypic and functional effects, and conducting gene expression studies of evolved isolates. Consistent with the complexity of solvent tolerance, we observe adaptations in diverse cellular processes. We find evidence of parallel evolution in *marC*, *hfq*, *mdh*, *acrAB*, *gatzABCD*, and *rph* genes. Many isobutanol tolerant lineages show reduced RpoS activity, likely related to mutations in *hfq* or *acrAB*. The first five mutations (in genes *marC*, *miaA-hfq*, *rph*, *mdh*, and *groL*) acquired in one lineage were reconstructed singly and in various combinations, revealing negative epistasis between *hfq* and *marC*, but predominantly positive epistasis between either of *hfq* or *marC* and subsequent mutations. These results provide an interesting supplement to recent reports of prevalent negative epistasis between beneficial mutations (Khan, et al. 2011; Chou, et al. 2011). Collectively, our results suggest mechanisms of adaptation to isobutanol stress based on remodeling the cell envelope and surprisingly, stress response attenuation.

Through evolution and genome re-sequencing work, we ultimately identified 247 genetic loci potentially associated with

isobutanol tolerance. We are currently performing targeted mutagenesis on a select subset of 38 genetic loci using Multiplex Automated Genome Engineering (MAGE), a recently developed technology entailing repeated cycles of high efficiency recombination using libraries of mutagenic DNA oligonucleotides (Wang, et al. 2009). This strategy enables rapid exploration of vast genotype space without being constrained to adaptive walks. Variants with improved isobutanol tolerance will be isolated using high-throughput phenotype screening with a microfluidic platform, then subjected to further genotype and phenotype characterization. This allows large-scale systematic correlation of isobutanol tolerance phenotypes and genotypes, yielding additional insights into mechanisms of tolerance, as well as generating improved strains of *E. coli* for isobutanol production.

References

- Minty, J. J., Lesnfsky, A. A., Lin, F., Chen, Y., Zaroff, T. A., Veloso, A. B., Xie, B., McConnell, C. A., Ward, R. J., Schwartz, D. R., Rouillard, J., Gao, Y., and Lin, X. N. (2011). Evolution combined with genomic study elucidates genetic bases of isobutanol tolerance in *Escherichia coli*. *Microbial Cell Factories*, **10**:18.
- Khan, A. I., Dinh, D. M., Schneider, D., Lenski, R. E., and Cooper, T. F. (2011). Negative epistasis between beneficial mutations in an evolving bacterial population. *Science*, 332:1193-1196.
- Chou, H., Chiu, H., Delaney, N. F., Segré, D., and Marx, C. J. (2011). Diminishing returns epistasis among beneficial mutations decelerates adaptation. *Science*, 332:1190-1192.
- Wang, H. H., Isaacs, F. J., Carr, P. A., Sun, Z. Z., Xu, G., Forest, C. R., and Church, G. M. (2009). Programming cells by multiplex genome engineering and accelerated evolution. *Nature*, 460:894-898.

Statistical Analysis of Liposome Budding Dynamics Based on Free Energy Landscape

Soichiro Tsuda¹, Hiroaki Suzuki^{1,2} and Tetsuya Yomo^{1,2,3}

¹Yomo Dynamical Micro-scale Reaction Environment Project, ERATO, Japan Science Technology Agency

²Graduate School of Information Science and Technology, Osaka University

³Graduate School of Frontier Biosciences, Osaka University

yomo@ist.osaka-u.ac.jp

Extended Abstract

As early Artificial Life research pointed out (Varela, et al.1974, Zeleny, 1977), the function of boundary structures encapsulating self-replicating information materials has been increasingly important in artificial cell (protocell) research. It has been speculated that there are two key factors for synthesizing self-replicating protocells: informational carrier and membrane compartment (Szostak, 2001). Now that technologies are catching up to implement "wet artificial life", the minimal condition for self-replication of informational materials (e.g. DNA) has been nailed down to a certain level. For example, a cell-free protein synthesis system from all well-known materials was developed (Shimizu, et al. 2001), and several research groups have constructed systems that self-replicate informational materials using biological components (Ichihashi, et al. 2010, Oberholzer et al. 1995, Rasmussen, et al. 2003). Then, the next question would be to demystify conditions for the self-replication of membrane compartments.

Of particular interest here is the shape transformation dynamics of lipid bilayer vesicles (liposomes), a commonly-used encapsulating material for chemical reaction system (Luisi and Stano, 2010). It has been long known that liposomes change shapes upon the addition of external stimuli, such as osmotic pressure. In some cases they form buds and eventually develop daughter vesicles, just like cell division. Inspired by conformation analysis of protein folding dynamics (Maisuradze, 2009), we attempt to capture a broad picture of this shape transformation (budding) dynamics of liposome by reconstructing effective potential energy landscape (FEL).

The experiment and analysis were carried out as follows: First, giant unilamellar liposomes were exposed to a hypertonic solution and forced to change shapes due to osmotic pressure. Fluorescent cross-sectional images of liposomes were then taken by a confocal microscope. By image analyses, we have measured perimeter, area, the longest and shortest length, equivalent diameter (the diameter of a circle with the equal area), the area of convex hull, and eccentricity of each liposome. Based on these measures, five shape-characterising indexes, Elongation, Area-perimeter, Eccentricity, Solidity, Roundness Factor, are calculated. This five-dimensional data set was mapped onto lower dimensions (1D or 2D) by principal component analysis, while keeping the original structures in the 5D space. Finally, an effective FEL can be reconstructed as $F = -k_B T \log(P/P_{\max})$, where P is the probability distribution function obtained from the frequency distribution in the principal component space and P_{\max} is the maximum probability in P .

Figure 1 (right) shows an example of reconstructed FELs. The landscape extends towards two directions: upper right (near (4)) and lower right (near (2)), which correspond to vesicles of bacteria-like and red blood cell-like shapes, respectively. Thus, the first and second principal components are effectively indexes for outward and inward deformations, respectively. The landscape has two local minima around ① and ④, which correspond to spherical and spherocylindrical liposomes, respectively. From a separate experiment, we obtained temporal snapshot images of a single budding liposome (Fig.1 left, Tsuda, et al. 2012). When plotted the sequential images onto the FEL as a trajectory, it was found that the behaviour of budding liposome was consistent with the configuration of the FEL, i.e. a fluctuating spherical liposome stays around at one of local minima for a certain period (①), and then jump to another local minimum (④) via a transition state (③). The vesicle eventually formed a bud and developed a small daughter vesicle. Temporal behaviour of liposomes in other cases also agreed with reconstructed FELs.

Theoretical models of vesicle shape transformation well explain shape transformations under defined parameters, reduced volume (V_{red}) and area difference between inner and outer monolayers (ΔA). However, liposomes in a population may well have the heterogeneity regarding these parameters and it is very difficult to measure these parameters experimentally (particularly the latter).

Accordingly, shape transformations of lipid vesicles can be very diverse in actual experiments. The budding transformation of liposome described above is in fact a rare event and difficult to capture when one tracks temporal behaviour of a single liposome. The FEL analysis provides information on possible shapes of liposome under a specific condition as well as allows us to predict rare transformations. As it can be applied to any shape transformations, this analytical method can be a handy tool to estimate the dynamics of any protocell models potentially possessing life-like cell-division dynamics.

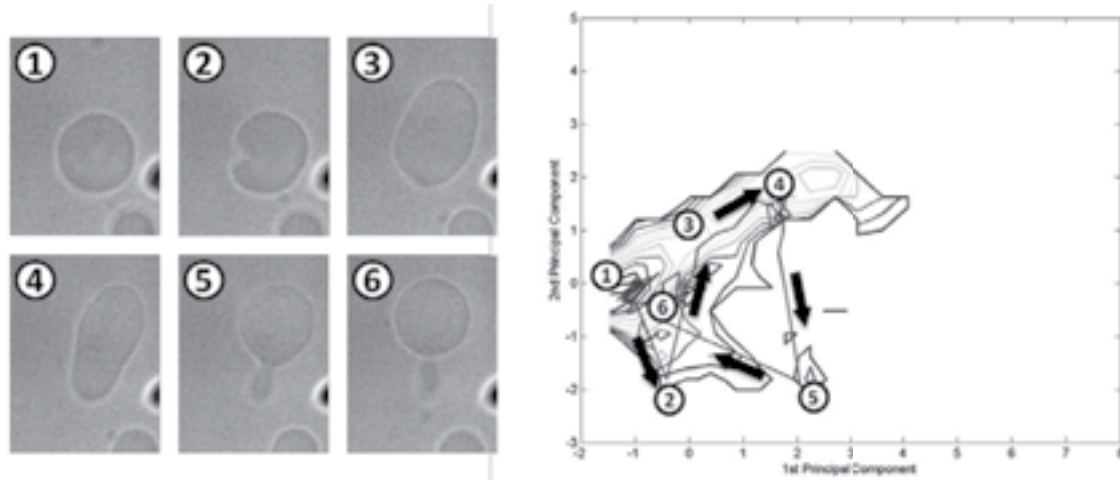


Figure 1: Snapshots of a budding liposome (left) and the corresponding positions in a reconstructed FEL (right).

References

- Varela, F. J., Maturana, H. R., and Uribe, R. (1974). Autopoiesis: The Organization of Living Systems. *Biosystems*, 5(4):187-196.
- Zeleny, M. (1977). Self-Organization of Living Systems: A Formal Model of Autopoiesis. *International Journal of General Systems*, 4(1):13-28
- Szostak, J. W., Bartel, D. P., and Luisi, P. L. (2001). Synthesizing life. *Nature*, 409(6818):387-90.
- Shimizu, Y., Inoue, a, Tomari, Y., Suzuki, T., Yokogawa, T., Nishikawa, K., and Ueda, T. (2001). Cell-free translation reconstituted with purified components. *Nature biotechnology*, 19(8):751-755.
- Ichihashi, N., Matsuura, T., Kita, H., Sunami, T., Suzuki, H., and Yomo, T. (2010) Constructing partial models of cells, *Cold Spring Harb Perspect Biol* 2, a004945.
- Oberholzer, T., Albrizio, M., and Luisi, P. L. (1995). Polymerase chain reaction in liposomes. *Chemistry and biology*, 2(10): 677-82.
- Rasmussen, S., Chen, L., Nilsson, M., and Abe, S. (2003). Bridging nonliving and living matter. *Artificial life*, 9(3): 269-316.
- Luisi, P. L., and Stano P., (2010) *The Minimal Cell: The Biophysics of Cell Compartment and the Origin of Cell Functionality*, Springer.
- Maisuradze, G. G., Liwo, A., and Scheraga, H. A. (2009). Principal component analysis for protein folding dynamics. *Journal of Molecular Biology*, 385(1): 312-329.
- Tsuda, S., Suzuki, H., and Yomo, T. (2012) A supplementary movie for this is available at <http://youtu.be/c8SvRIC2els>

The Origin of Life is a Spatially Localized Stochastic Transition

Meng Wu and Paul G Higgs

Origins Institute and Dept. of Physics and Astronomy,
McMaster University, Hamilton, Ontario L8S 4M1, Canada.
higgsp@mcmaster.ca

Extended Abstract

The creation of an autocatalytic reaction system controlled by polymers such as RNA is the key step in the origin of life. We have previously studied scenarios for the origin of the RNA World (Wu and Higgs, 2008, 2010) using mathematical models of RNA polymerization. These models have two stationary states. In the non-living state, polymerization is possible to some degree at a slow spontaneous rate, but the system is dominated by monomers and short oligomers. In the living state, reaction rates are controlled by ribozymes and there is a significant concentration of long polymers. In a large, well-mixed system, the non-living state is dynamically stable indefinitely. However, in a finite sized region, with finite numbers of molecules, concentration fluctuations can cause a stochastic transition from the non-living to the living state. Here, we consider a simplified generic model of replicators that has the same essential features as our RNA polymerization models. This allows us to investigate the effect of the spatial distribution of replicators on the stochastic transition that leads to the origin of life.

We consider a system with N replicators on a lattice with a constraint that no more than three replicators may be present per site. The density, relative to the carrying capacity, is ϕ . Replicators can appear at rate s , representing a slow rate of synthesis by random polymerization. Existing replicators may be copied at a rate r , representing a process of non-living template-directed synthesis. Replicators may also act as polymerases that catalyze the replication of another replicator at rate k . The latter process occurs when there are exactly two replicators on a site - "two's company, three's a crowd". Replicators also die at a rate 1 and can hop to neighbouring sites at a rate h . When h is large, the system is well-mixed, and the dynamics obeys the equation:

$$d\phi/dt = (s + r\phi + k\phi^2)(1 - \phi) - \phi$$

The stationary states are the roots of this cubic equation. The two stable states, ϕ_1 and ϕ_2 , are separated by an unstable state ϕ_3 . If the system is well-mixed, it remains in the dead state at density ϕ_1 . When h is lower, the density converges initially to ϕ_1 , but then a stochastic transition occurs, leading to a high density in a localized region. The high density patch then spreads deterministically across the lattice (see Fig. 2) and the density increases to the living state, ϕ_2 .

The time required for the origin of life depends on the lattice size and on the diffusion rate h . In the well-mixed limit, the transition requires global-scale concentration fluctuations and becomes increasingly more difficult as the lattice size increases. In the low- h case, widely-separated regions behave independently; hence the time required decreases with lattice size. This model illustrates that life arises by a rare stochastic event that occurs due to spatially localized concentration fluctuations. Once the living state is established locally, it can spread deterministically through the rest of the system. These are generic features also possessed by more complex models with a greater degree of chemical realism.

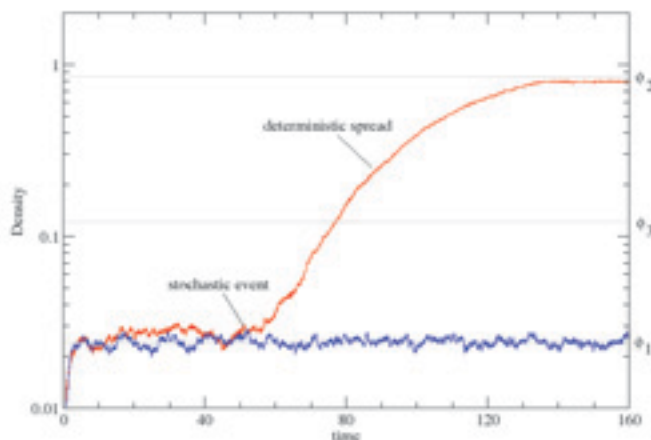


Figure 1. Blue curve - a typical simulation in the well-mixed regime remains stable in the non-living state. Red curve - a typical simulation with low h goes through a stochastic transition from the non-living to the living state.

References

- Wu M, Higgs PG (2009) Origin of Self-replicating Biopolymers: Autocatalytic Feedback can Jump-start the RNA World. *J. Mol. Evol.* 69: 541-554.
 Wu M, Higgs PG (2011) Comparison of the roles of nucleotide synthesis, polymerization and recombination in the origin of autocatalytic sets of RNAs. *Astrobiology* 11: 895-906.

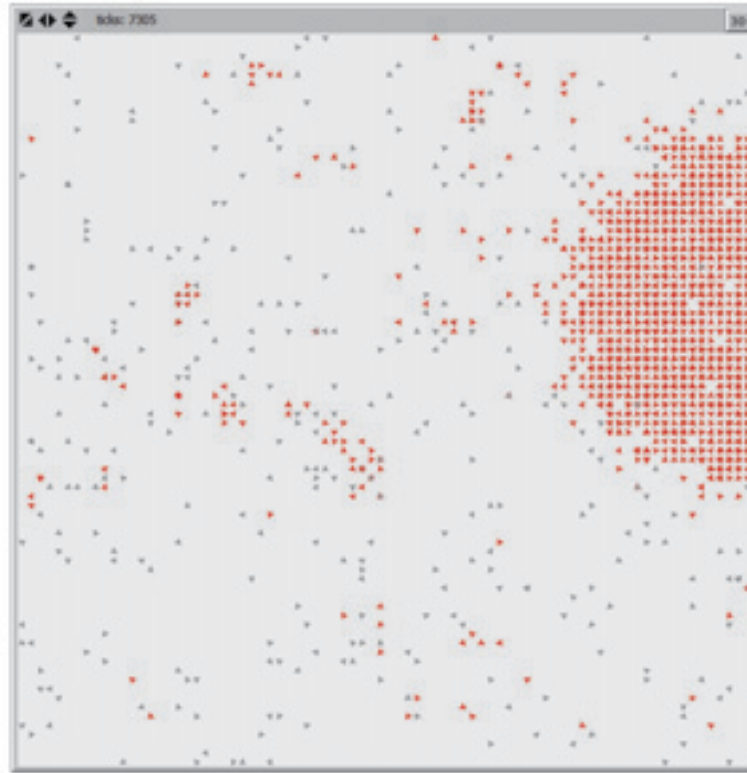


Figure 2. Snapshot of a system shortly after the transition to life. The non-living state is characterized by a low density of replicators created by spontaneous synthesis (coloured grey). The living state state is a dense patch of replicators that have been synthesized catalytically (red). Once it is big enough to be stable, the living patch spreads deterministically across the lattice.

The Humanities and ALife

Extended Abstracts

Automated Evolution of Interesting Images

Joshua E. Auerbach¹

¹Morphology, Evolution & Cognition Laboratory
Department of Computer Science, University of Vermont
Burlington, VT 05405
joshua.auerbach@uvm.edu

Extended Abstract

Recent work (Secretan et al., 2011) has demonstrated that it is possible to evolve interesting images produced by Compositional Pattern Producing Networks (CPPNs) (Stanley, 2007) through interactive evolution. However, interactive evolution is a slow process that requires the active involvement of human users. It is desirable to evolve interesting images without requiring human users to perform selection. In this work I explore alternate methods of evolving interesting images from CPPNs that are completely automated, yet in some cases still indirectly informed by what humans find interesting.

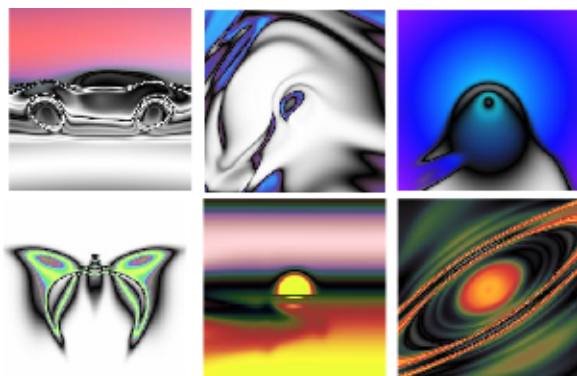


Figure 1: Sampling of interesting and familiar looking images produced by Picbreeder. Taken from (Secretan et al., 2011).

Picbreeder (<http://www.picbreeder.org>) is a website created by researchers in the EPLEX group at the University of Central Florida for the purpose of interactively evolving images. This system has produced many interesting and familiar looking images. A small sampling of these can be seen in Figure 1. The images in Picbreeder are produced by CPPNs, a form of indirect encoding that abstracts biological development to produce outputs with the familiar biological properties of symmetry, repetition and repetition with variation. Commonly CPPNs are evolved via CPPN-NEAT, an extension of the state of the art NeuroEvolution of Augmenting Topologies (NEAT) (Stanley and Mi-

ikkulainen, 2001) algorithm, applied to CPPNs. Picbreeder employs this algorithm with selection based on the interactive choices of human users over the internet. Likewise, this algorithm (or variants of it) are employed in the work presented here¹.

In lieu of employing human users in the experiments conducted in this work, the fitness of individual images is calculated automatically and selection is based on these calculated fitness values. 400×400 pixel images are evolved. The two primary ways in which these images' fitnesses are evaluated are their complexity, defined in terms of the size of a zlib compressed representation of the image and their ability to maximize the number of results returned when the image is used as a search query to Google's search by image (SBI) feature².

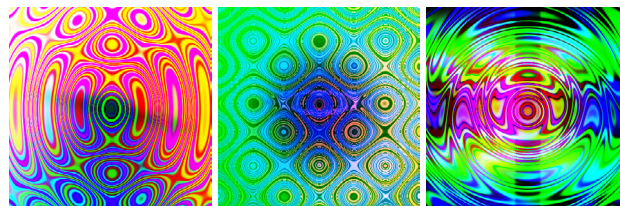


Figure 2: Sampling of images evolved to be maximally complex. These images evoke 1960s concert posters.

The initial hope was that by maximizing the number of results when querying SBI it would be possible to indirectly capture notions of interestingness, because images which are interesting to humans are precisely those likely to exist on the internet and be indexed by Google. However, when the sole objective of evolution was to maximize this number evolution tended to converge on images that were entirely one color (though the specific color varied across evolutionary trials). It is hypothesized that this is

¹The selection mechanism is what differentiates the current work from Picbreeder. The CPPNs evolved in this work make use of the same inputs and outputs as those in Picbreeder including the use of the Hue Saturation Brightness (HSB) color space.

²<http://www.google.com/imghp?sbi=1>

because searching on a single color will find many images containing that color, but the details of SBI are proprietary and unknown to the author.

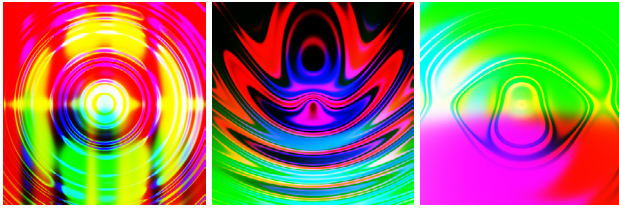


Figure 3: Sampling of images evolved to maximize complexity and SBI hits combined multiplicatively into a single fitness function.

Alternatively, complex images tend to be interesting so by selecting for maximally complex images it should be possible to produce interesting results. A sampling of images selected to be maximally complex is shown in Figure 2. Finally, by combining these objectives either multiplicatively or through pareto based multi-objective selection³ it should be possible to evolve images both complex and informed by the corpus of images that exist on the web. Images evolved under both these schemes are shown in Figures 3 and 4 respectively.

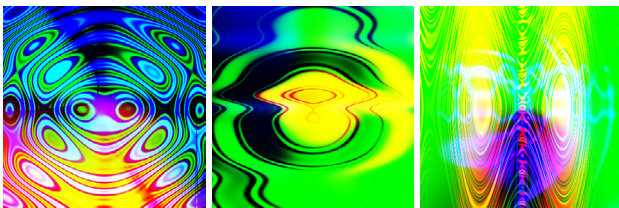


Figure 4: Sampling of images evolved to maximize complexity and SBI hits using Pareto based multi-objective optimization.

While none of these selection mechanisms produced images with the familiarity of those produced by Picbreeder the results are still interesting to look at for artistic purposes. Modifying the evolutionary search in other ways, such as by allowing recurrent connections within the CPPN genomes may allow for the creation of images that are interesting in distinct ways and is beginning to be investigated (see Figure 5 for examples). Additionally it is possible that by using a larger number of evaluations⁴, by combining these fitness criteria with others, or by incorporating keywords into the

³For this purpose CPPN-NEAT was modified to use a selection mechanism based on NSGA-II (Deb et al., 2002)

⁴Here the number of evaluations was rate limited because querying Google too frequently will cause one's IP address to be blocked.

SBI searches it may be possible to automatically evolve images that are both interesting and familiar without resorting to the direct user evaluations employed in Picbreeder.

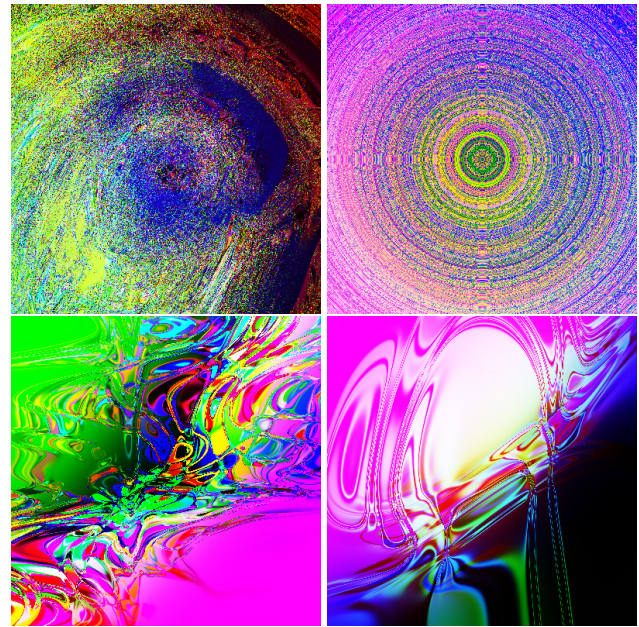


Figure 5: Sampling of images evolved using recurrent CPPN connections.

Acknowledgements

This work was supported by National Science Foundation Grant PECASE-0953837.

The author would like to thank Peter Andrews of Dartmouth College for a conversation that led to the development of the ideas in this paper.

The author also acknowledges the Vermont Advanced Computing Core which is supported by NASA (NNX 06AC88G), at the University of Vermont for providing High Performance Computing resources that have contributed to the research results reported within this paper.

References

- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6:182–197.
- Secretan, J., Beato, N., D'Ambrosio, D. B., Rodriguez, A., Campbell, A., Folsom-Kovarik, J. T., and Stanley, K. O. (2011). Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation Journal*, pages 373–403.
- Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162.
- Stanley, K. O. and Miikkulainen, R. (2001). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:2002.

EndlessForms.com: Collaboratively Evolving 3D-Printable Objects Online

Jeff Clune*, Jason Yosinski*, Eugene Doan, and Hod Lipson

Cornell University, 239 Upson Hall, Ithaca, NY 14853, USA. *Equal contributors.
jeffclune@cornell.edu

Extended Abstract

Overview

This abstract introduces EndlessForms.com, the first website to allow users to interactively evolve three-dimensional (3D) shapes online. Visitors are able to evolve objects that resemble natural organisms and engineered designs because the site utilizes a generative encoding inspired by concepts from developmental biology (Figure 1). This Compositional Pattern Producing Network (CPPN) encoding abstracts how natural organisms grow from a single cell to complex morphologies. With the click of a button, visitors can 3D print each evolved object in materials ranging from plastic to silver. The site takes advantage of a new Web technology technology called WebGL that enables the visualization of 3D objects in Internet browsers. EndlessForms.com thus brings together recent innovations in evolutionary computation, Web technologies, and 3D printing to create a powerful collaborative interactive evolution experience that was not possible until 2011. Since going live in that year, over 3 million objects have been evaluated on the site during more than 200,000 generations of interactive evolution. A sizable community of citizen scientist participated: there were over 40,000 unique visitors from 150 countries and all 50 US states. In addition to its scientific mission of fueling intuitions regarding generative encodings for evolutionary algorithms, EndlessForms serves an educational outreach goal: visitors learn about evolution and developmental biology in a fun virtual setting and can transfer the 3D objects they create to the physical world (Figure 2).

View a video tour of EndlessForms at <http://goo.gl/YvoBw>

Compositional Pattern Producing Networks (CPPNs) abstract the process of natural development without simulating the low-level chemical dynamics involved in developmental biology (Stanley, 2007). Cells (and higher-level modules) in natural organisms often differentiate into their possible types (e.g. heart or spleen) as a function of where they are situated in geometric space (Wolpert and Tickle, 2010). With CPPNs, phenotypic elements are similarly specified as a function of their geometric location (Stanley, 2007). Each CPPN is a directed graph in which every node is itself a single function, such as sine or Gaussian. The nature of the functions can create a wide variety of desirable properties, such as symmetry (e.g. a Gaussian function) and repetition (e.g. a sine function) that evolution can exploit (Figure 1).

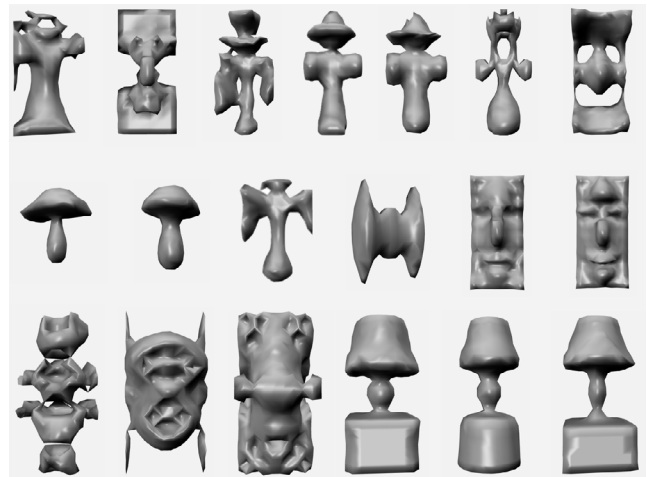


Figure 1: Example objects evolved on EndlessForms.com. Because they are evolved with a generative encoding based on developmental biology, the objects exhibit important properties seen in natural and engineered designs, such as symmetry and repetition, with and without variation.

To evolve 3D objects, inputs for the x , y , and z dimensions, and the distance from center, are provided to a CPPN. A *workspace* (maximum object size) is defined with a *resolution*, which determines the number of voxels in each dimension. On EndlessForms.com there are 10 voxels in the x and z dimensions and 20 in the y (vertical) dimension. These four values are iteratively input to a CPPN, and voxels are considered full if the CPPN output is greater than a threshold (here set to 0.1), otherwise the voxel is considered empty. The 3D voxel array is then processed by the surface-smoothing Marching Cubes algorithm. A normal is provided for each vertex when visualizing the objects in WebGL, which allows the renderer to further smooth the surface. These two smoothing steps enable high-resolution CPPN objects to be visualized without prohibitive computational costs.

3D objects are evolved with interactive evolution. The website user views 15 rotating objects and selects the parents



Figure 2: Objects evolved on EndlessForms printed in plastic, silver, and bronze. A “3D Print” button on the page for each object sends the design to our 3D printing partner Shapeways.com.

of the next generation. The algorithm and its parameters are described in Clune and Lipson (2011).

Crowdsourced evolution has been previously implemented by websites like Picbreeder.org, which allows users to evolve 2D images with CPPNs (Secretan, 2011). The complexity and natural appearance of the resulting images often support claims regarding the legitimacy of CPPNs as an abstraction of biological development (Stanley, 2007). It is possible that CPPNs are unable frequently to make sensible forms with the added difficulty of another dimension, and when objects must be one contiguous unit (which aids in transfers to reality). Thus, a demonstration in 3D significantly strengthens these claims of legitimacy, because the natural world is three-dimensional. Evolving CPPN objects in the natural 3D setting demonstrates that generative encodings based on *geometric* abstractions of development capture some of the complexity-generating power of natural morphological development. Doing so also provides a visually intuitive testbed for studying how variants of such generative encodings behave. It also reveals the utility of CPPNs as a representation for 3D object design (Clune and Lipson, 2011).

We previously described how to evolve 3D shapes with CPPNs on a personal computer (Clune and Lipson, 2011), and 3D objects have been evolved in hardware (Rieffel and Sayles, 2010). However, crowdsourcing represents a fundamentally different way of exploring a design space (Secretan, 2011). By allowing visitors to share designs and further evolve them, innovations discovered by one user can be built upon by the crowds that follow. For example, once a user found a mushroom or lamp design, other users generated many interesting variants on that theme (Figure 1). There

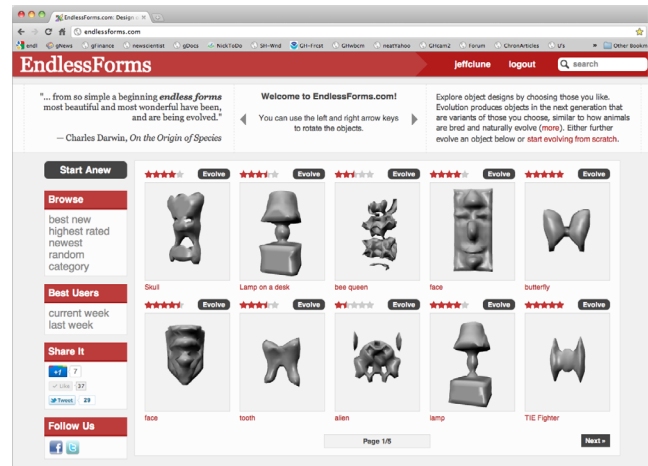


Figure 3: The EndlessForms.com homepage, including objects that users have evolved and rated highly.

are currently over 60 lamps and 20 mushrooms descendent from those two discoveries. To catalyze such crowdsourced efforts we enabled users to share their discoveries via Facebook and Twitter (Figure 3). Collaborative evolution also means that no individual has to perform all the evaluations between generation 0 and a new discovery, facilitating deep searches into promising areas of the search space. Users can further evolve objects published by others, or start anew from randomly generated genomes, which increases diversity in the search. Evolved objects can be brought into the physical world via 3D printing (Figure 2), which creates a fun incentive for users to keep evolving.

In our presentation we will describe the results of this experiment in collaborative object design, including the types of objects evolved and the effect that crowdsourced evolution had on the exploration of the design space. We will also discuss future directions for harnessing crowds to facilitate research in interactive evolution, generative encodings, and automated object design.

References

- Clune, J. and Lipson, H. (2011). Evolving three-dimensional objects with a generative encoding inspired by developmental biology. In *Proc. European Conf. Artificial Life*, pages 144–148.
- Rieffel, J. and Sayles, D. (2010). Evofab: a fully embodied evolutionary fabricator. In *Proc. ICES*, pages 372–380.
- Secretan, J et al. (2011). Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, 19(3):373–403.
- Stanley, K. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162.
- Wolpert, L. and Tickle, C. (2010). *Principles of Development*. Oxford University Press, 4th edition.

Germs, Genes, and Memes: Function and Fitness Dynamics on Information Networks

Patrick Grim,^{1,4} Daniel J. Singer,² Christopher Reade,³ Steven Fisher⁴

¹Department of Philosophy, Stony Brook University, Stony Brook, NY 11794

²Department of Philosophy, University of Michigan, Ann Arbor, MI 48109

³Gerald R. Ford School of Public Policy, University of Michigan, Ann Arbor, MI 48109

⁴Center for Study of Complex Systems, University of Michigan, Ann Arbor, MI 48109
pgrim@notes.cc.sunysb.edu

Extended Abstract

Genes, germs, and memes are all forms of information transfer across networks. What are the differences in network dynamics between them? What are the differences in fitness or functionality? Concentrating on the specific case of transfer between sub-networks, we compare both the dynamics of each of these across networks and on their comparative fitness. We focus on both (a) types of sub-networks and (b) the degree of linkage between them (Fig. 1). For each form of information transfer, we compare increased linkage between sub-networks of a specific type with the same increased linkage within a single network of that type (see also Grim, Reade, Singer, Fisher, & Majewicz 2010; Golub & Jackson, forthcoming).

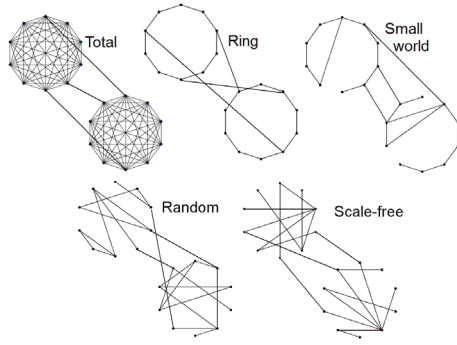


Fig. 1 Network types at issue.

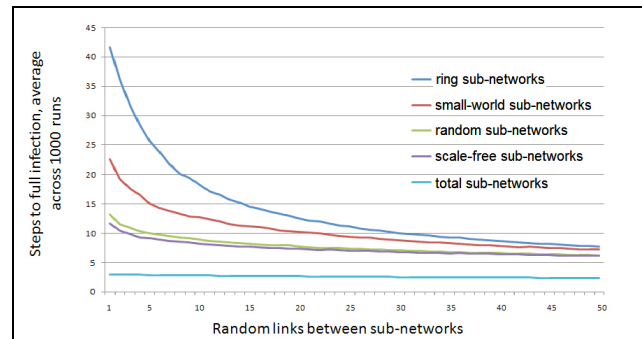


Fig. 2 Average time to total infection with increasing links between sub-networks

Germs and memes, it turns out, show a very different network dynamics. For infection, measured in terms of time to total infection, it is network type rather than degree of linkage between sub-networks that is of primary importance. Fig. 2 shows the average time to total infection for different network types, each of which is virtually identical to results for a single network of that type.

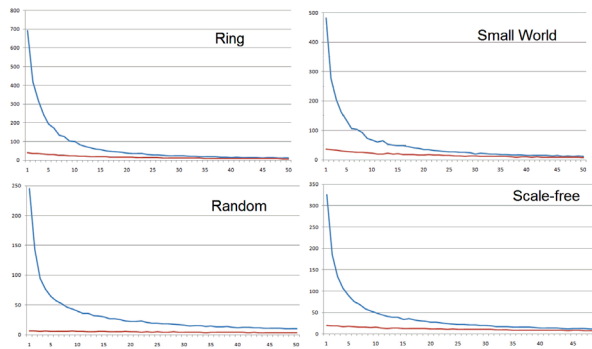


Fig. 3 Times to belief convergence for increasing links between sub-networks (blue) and within single networks (red).

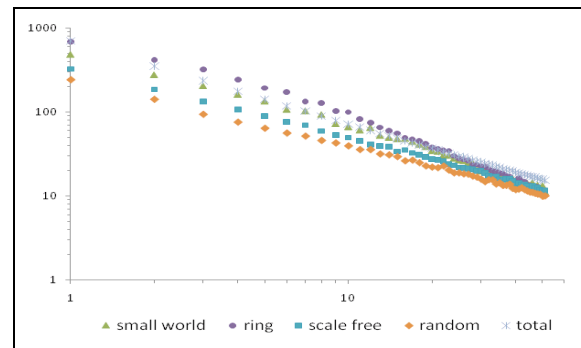


Fig. 4 Log-log plots of times to belief consensus with increased linkage between sub-networks

We model belief transfer by assigning an initial belief value between 0 and 1 to each of our agents, updating by averaging with network contacts. Reinforcement dynamics of this type gives results precisely opposite to the case of infection. In the case of infection, network type trumped degree of linkage. For belief transfer, measured in terms of time to consensus, it is degree of linkage that trumps network type. Figure 3 shows the clear difference between single and linked sub-network results for each network type. Figure 4 shows parallel results for increased linkages between sub-networks regardless of network type.

Transfer of genetic information, modeled by cross-over, matches neither of the other dynamics in full but does exhibit intriguing features of each. Like infection but unlike belief, genetic transfer shows little difference between the case of single and linked sub-networks (Fig. 5). Like belief but unlike infection, however, network type makes very little difference to genetic transfer. Here, as in the case of belief, we have the signature of a power law (Fig. 6).

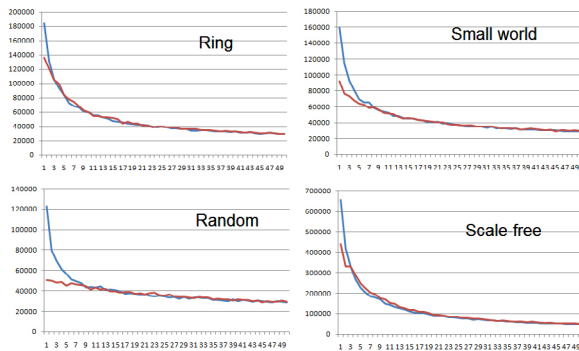


Fig. 5 Times to genetic convergence for increasing links between sub-networks (blue) and within single networks of that type (red).

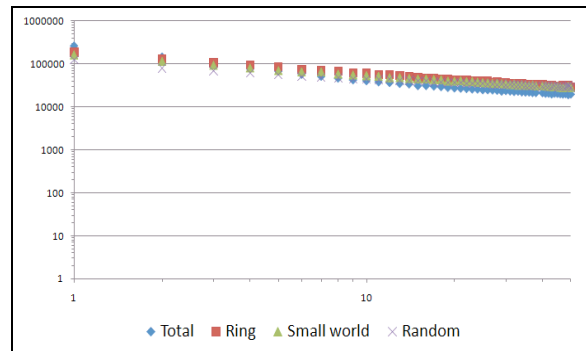


Fig. 6 Log-log plots of times to genetic consensus with increased linkage between sub-networks

In order to compare fitness we developed a uniform way of coding information in each case and measured progressive approximation to a specific code most strongly selected for. Within such a framework we could compare genes, germs, and memes in terms of both speed and final proximity to the optimal code. Figure 7 shows a log plotting of relative speeds for a single scale-free network, typical of relative speeds across all cases. Figure 8 shows a log of relative fitness at convergence for germs, genes, and memes. Belief reinforcement proves the most fit, the asexual reproduction of infection the least, with the fitness of genetic information transfer somewhere between the other two.

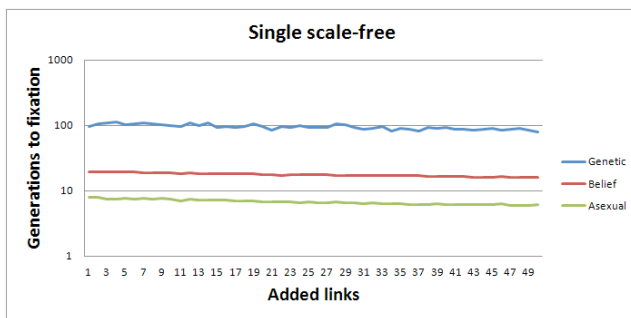


Fig. 7 Relative speed to convergence for information types

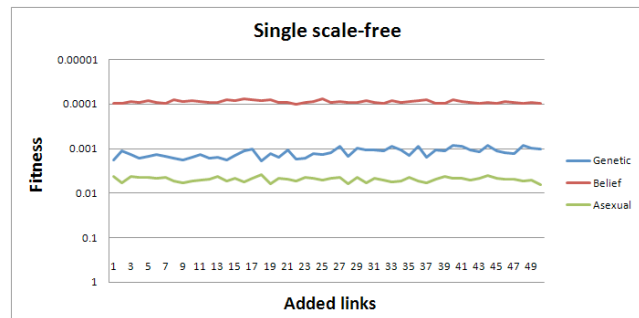


Fig. 8 Relative fitness for information types

Each of the information transfer mechanisms considered displays both a specific dynamics and a specific fitness pattern across network features. Information transfer is not all of a kind; attention to differences will be important in understanding both its natural instantiations and in employing those artificially.

References

- Golub, B., & Jackson, M. O. (forthcoming). How homophily affects learning and diffusion in networks. Available as a working paper at www.bepress.com/feem/paper296
- Grim, P., Reade, C., Singer, D. J., Fisher, S. and Majewicz, S. (2010). What you believe travels differently: Information and infection dynamics across sub-networks," *Connections* 30, 50-63.

Biology of Digital Organisms: How Language Constructs Reality

Orly Kramash Stettiner¹

¹Program in Science, Technology & Society (STS), Bar-Ilan University, Israel
orlyst@netvision.net.il

Extended Abstract

When Christopher Langton first coined the term "artificial life" and organized the first conference of the nascent field, he envisioned that "We would like to build models that are so life-like that they cease to become models of life and become examples of life themselves." (Langton 1989). A few years later, the American postmodern literary critic Katherine Hayles was startled by this vision and by Thomas Ray's reference to his Tierra creatures as "natural forms of life", and wondered how it was possible, in the late twentieth century, to "believe, or at least claim to believe, that computer codes are alive? And not only alive, but natural?" (Hayles 1996). In her opinion, the explanation stems not from the scientific context of the programs but from the stories told about and through them, multilayered systems of metaphors, visual representations and redefinitions of basic concepts.

The American philosopher of science Evelyn Fox Keller supported Hayles's view and generalized it into the linguistic domain. She claimed that the ALife community developed an extensive biological lexicon for interpreting their models (Keller 2002). This lexicon, she wrote, "adds substantively to the sense of proximity to the real-life examples for which they aim" (p. 277). Fox Keller attributes less significance to the question of whether these artificial creatures are alive or not, claiming that such a distinction is merely a matter of human categorization that does not carry any practical implications. She emphasizes the increasingly narrowing and illusory gap between computers and organisms, as reflected by the terms "computational biology" and "biological computation", wondering if this convergence (both material and conceptual) leads to an indistinguishable gap between the living and non-living.

This contribution is intended to revisit Hayles and Fox Keller, examining how the usage of language and visualization tools have continued to construct and shape the field of ALife in the decade since their articles were published.

Through an examination of leading research reports in the field of digital evolution (e.g. (Lenski, Ofria et al. 2003), (Hazen, Griffin et al. 2007), (McKinley, Cheng et al. 2008) and others), we demonstrate an extensive and seemingly deliberate usage of re-defined biological concepts and anthropomorphisms of digital organisms, creating a new mainstream vocabulary, in which a certain piece of code present in the computer memory becomes a living organism and a unique sequence of instructions constructs a genotype. We employ the thinking of the Swiss linguist Ferdinand de Saussure and the French philosopher Jacques Derrida, who recognized the importance of language as a tool for reality construction.

The expanding usage of biological vocabulary (Figure 1) illustrates a deliberate affiliation and self-identification on the part of ALife researchers with the discipline of biology, rather than that of computer science. This claim is supported by the increasing reliance of digital evolution studies on scientific publications and lab reports written by biologists (and vice versa), which has the effect of conceptually fixing the notion of the resemblance between digital computer simulation results and in-vivo laboratory experimental results. Special attention is given to the visual presentation of digital organisms, which has a central role in creating a vivid, living conception of these evolving pieces of code. In addition, we recognize the extensive usage of methodologies, research and analysis tools adopted from molecular biology, resulting in the intensification of the illusion of similarity and even identity between living and artificial species.

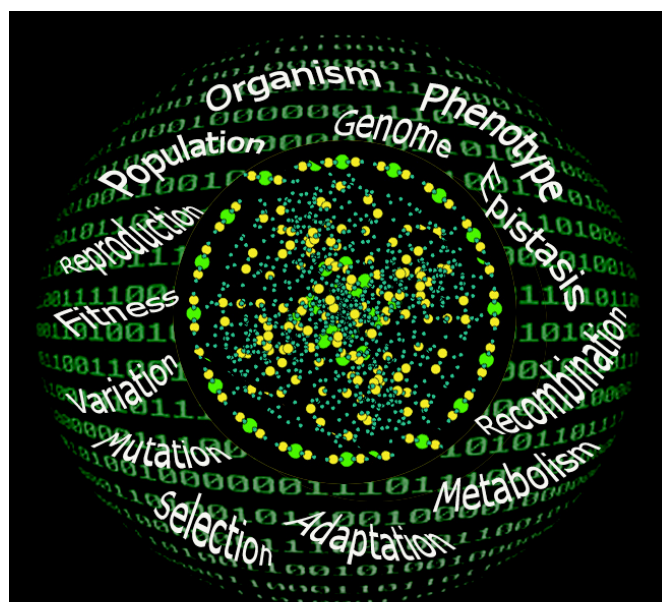


Figure 1: An illustration of some biological concepts being redefined by ALife researchers.

References

- Hayles, K. (1996). Narratives of Artificial Life. *Futurenatural: Nature, Science, Culture*. M. M. George Robertson, Lisa Tickner, Jon Bird, Barry Curtis and Tim Putnam (editors). London, Routledge: 147-164.
- Hazen, R. M., P. L. Griffin, et al. (2007). "Functional information and the emergence of biocomplexity." *PNAS (Proc. National Academy of Science)* **104**(Suppl. 1): 8574-8581.
- Keller, E. F. (2002). *Making Sense of Life: Explaining Biological Development with Models, Metaphors and Machines*. Cambridge, Massachusetts, Harvard University Press.
- Langton, C. G. (1989). Artificial Life- the proceedings of an interdisciplinary workshop on the synthesis and simulation of living systems. *Artificial Life*, Redwood City, CA, Addison-Wesley.
- Lenski, E. R., C. Ofria, et al. (2003). "The evolutionary origin of complex features." *Nature* **423**: 139-144.
- McKinley, P., B. H. C. Cheng, et al. (2008). "Harnessing Digital Evolution." *IEEE Computer* **41**(1): 54-63.

The VIDA Art and Artificial Life Competition: Key Contributions to the Arts

Nell Tenhaaf¹, Mónica Bello Bugallo², with Sonia Cillari, Jose Carlos Mariátegui, Sally Jane Norman and Paul Vanouse

¹Professor, York University, Toronto, Canada
tenhaaf@yorku.ca

²Artistic Director, VIDA Art and Artificial Life International Competition
mb@monicabello.org

Extended Abstract

The annual VIDA Art and Artificial Life competition has become the preeminent international touchstone for Alife-inspired art. This paper describes some key themes that have emerged in VIDA, showing the distinctiveness and reach of Alife art. As examples of these themes, VIDA prizewinners and jury members discuss unique materialities and metaphors elicited by dynamic environments, and expression of behaviours in the realm of the quasi-living.

The VIDA competition was launched in 1999, and has always regarded Alife research advances as a key reference point. In the earliest competitions, jurors were very caught up in discussions of whether the relationships between the research and artistic poles of VIDA are metaphorical, or are stronger than metaphor. Sally Jane Norman, who was a juror in the first competition and in many subsequent iterations, has described Alife as an expanded theatre with human and lifelike actants – is this metaphor, or does it also encapsulate core Alife research questions? Manuel De Landa, also a VIDA 2.0 juror, said that Alife art is necessarily metaphorical because it is a cultural absorption and interpretation of the research. This challenging question has been threaded through VIDA since its inception, and has been met by different jury members in very different ways.

Simon Penny's 2009 article "Art and Artificial Life – a Primer" presents a concise historical overview of Alife art. Penny develops terms that are central to Alife art such as aesthetics of behaviour, machine creativity, and the "evolved aesthetic object." He contextualizes Alife art within a historical frame for Alife itself, looking back to its 19th and 20th century precursors – vitalism, cybernetics, fractals, chaos theory, etc.. He notes that the first Alife artworks predate the late 1980s appearance of Alife by decades e.g., Gordon Pask's 1953 *Musicolor* and Grey Walter's *Machina Speculatrix* tortoises Elmer and Elsie, 1948. Noteworthy pioneering Alife artworks include Penny's own *Petit Mal* autonomous robot (1989-2005), Karl Sims' *Evolved Virtual Creatures* (1994), Jane Prophet's online ecosystem *Technosphere* (1995), and Theo Jansen's *Animari* self-propelling beach animals (1990).

The VIDA competition was launched to recognize unique Alife-inspired artworks and the artists who make them. Whether they are a metaphor for Alife research principles, or a materialization of those principles, VIDA winning projects have been defined by core Alife concepts: they explore boundaries between the living and the non-living; they are concerned with synthesized properties of living and life-like systems, incorporating natural and artificial elements in their physical appearance; and they are representations of dynamic processes, responsive behaviours, and evolving ecologies. In 2010, Artificial Life XII reiterated its core mandate of "identifying and synthesizing the critical properties of living and life-like systems" and VIDA continues to affirm its links to this mandate. As well, the hybrid forms of the artistic proposals submitted to VIDA, and transformation of the discipline of Alife itself, have prompted VIDA to continuously broaden its reach. ECAL 2011 (the European Conference on Artificial Life) had the bold theme and title "Back to the Origins of Life", which carried on the mission of the 2009 edition to reflect on increasingly blurred boundaries between living and non-living processes. Over the years, VIDA has further opened up the competition to themes of bio-inspired artificial processes such as synthesized cells, biological substrates for computation, and bio-engineering.

In September 2011, the authors (four jury members from the VIDA Art and Artificial Life annual competition, along with two VIDA prize-winners) presented a panel at ISEA2011 in Istanbul, Turkey.¹ The panelists wanted to give an overview of art projects and themes recognized by VIDA, how these reflect Artificial Life research, and how they have made an impact on the arts internationally. The panel brought together artists and cultural theorists who reflect on the fact that Alife art, like all art, is engaged with representation and affect, but with an added challenge that faces any area of knowledge engaged with dynamic processes: how to capture and persuasively express the dynamics of the continuously variable. Sonia Cillari discussed her installations that propose behaviours for experiencing specific kinds of hybrid spaces, behaviours not simply based on input-output (sensor-to-

display) but more pertinently concerned with the co-performance of human and non-human actors.

Paul Vanouse's practice takes up questions about behaviours within biology, at what levels they occur and the fact that we can detect them only through artificial means. For example, Vanouse's works propose that DNA is a bridging "behaviour" between the living and the non-living. Thus ALife shakes up comfortable and common sense notions of life. Bacteria are living but our understanding of their behaviours is a model that operates in the realm of the quasi-living. Cillari also deals in the quasi-living, in that the extension of bodies into an analog-to-digital sensorial space makes a quasi-living space.

The demonstration of behaviours in ALife artworks is key to exploring elisions among living, non-living and quasi-living features. Behaviour is intrinsically embodied, and so it also links metaphor with material instantiation. This behaviour-based materiality builds a scaffolding that gives distinctiveness – a particular shape and context – to ALife art practices within the broad sweep of international media art. In significant ways, VIDA's themes also link to broader social issues. Jose Carlos Mariátegui notes that projects submitted to the Incentive for New Productions, or "Incentivos" award inaugurated in 2001, often reveal artmaking cultures clearly connected to realms beyond the art world, much closer to needs and concerns such as development, sustainability and knowledge building. Essentially, the projects are more integrated into other communities than art practice generally is, and are linked to community building. Thus De Landa's remark about the metaphorical nature of ALife art may hold in certain ways, from the viewpoint of the established artworld; but from their own distinctive theoretical perspective, and in parallel with ALife research, ALife artworks are always grounded in the materiality of operating in a dynamic environment.

¹ ISEA is the International Symposium on Electronic Art. The panel was called VIDA: New Discourses, Tropes and Modes in Art and Artificial Life Research.

Acknowledgements

The VIDA Art and Artificial Life Competition is generously sponsored by Fundación Telefónica.

References

- Hayles, N. Katherine (1996). Narratives of Artificial Life. In Robertson, G., editor, *FutureNatural: Nature, Science, Culture*. Routledge, London.
- Penny, Simon (2009). Artificial Life Art – A Primer. In Penny, S., editor, *Proceedings of the Digital Arts and Culture Conference, 2009* at http://escholarship.org/uc/ace_dac09.
- Penny, Simon (1996). The Darwin Machine: Artificial Life and Interactive Art. *New Formations*, 29.

-
- Shanken, Edward (1998). Life as We Know It and/or Life as It Could Be: Epistemology and the Ontology/Ontogeny of Artificial Life. *Leonardo*, 31-5: 383-388.
- Tenhaaf, Nell (2008). Art Embodies A-Life: The VIDA Competition. *Leonardo*, 41-1: 6-15.
- Tenhaaf, Nell (2002). Perceptions of Self in Art and Intelligent Agents. In Dautenhahn, K., Bond, A.H., Cañamero, L., Edmonds, B. editors. *Socially Intelligent Agents: Creating Relationships with Computers and Robots*. Kluwer Academic Publishers, Norwell, MA.
- Tenhaaf, Nell (1998). As Art Is Lifelike: Evolution, Art, and the Readymade. *Leonardo*, 31-5: 397-404.
- Whitelaw, Mitchell (2004). *Metacreation: Art and Artificial Life*. The MIT Press, Cambridge.
- Whitelaw, Mitchell (1998). Tom Ray's Hammer: Emergence and Excess in A-Life Art. *Leonardo*, 31-5: 377-382.

A Self-sustaining Visual Feedback Machine using Chaotic Neural Dynamics

Alexander Woodward¹, Takashi Ikegami¹ and Yuta Ogai²

¹Ikegami Laboratory, Department of General Systems Studies, University of Tokyo, Tokyo, Japan.

²Department of Electronics and Mechatronics, Faculty of Engineering, Tokyo Polytechnic University, Tokyo, Japan.
alex@sacral.c.u-tokyo.ac.jp

Extended Abstract

We constructed a self-sustaining machine that generates video imagery, consisting of a camera, an internal mental visual feedback process, and a chaotic neural network with Hopfield structure and synaptic connections modified by Hebbian dynamics. Outside images and the internal visual feedback enter the neural network which in turn determines visual feedback parameters and camera control. The system does not always couple with the outside world and maintains its internal dynamics. Figure 1a gives an overview of the system design. This is an art installation but also a scientific work to investigate how an artificial life can be installed in the real world, developed as a minimal version of a previous work called Mind Time Machine (MTM), presented at the Yamaguchi Center for Media Art in 2010 (Ikegami, 2010).

The motivation behind this system involves a number of themes: How an organism sustains itself - here by maintaining internal dynamics even when uncoupled with the outside via the camera. How does a system's characteristic time (which we call Bergsonian time; the system's subjective time scale (Ikegami (2010))) emerge and how does it differ from Newtonian time? Here we consider the system's state update cycle as a candidate for Bergsonian time, with the update rate being a function of the neural state. How internal structures collaboratively work at different time scales; living systems organize their own time scales driven by their memory structures - our system has multiple time-scales: the neural update timescale, the memory accumulation process and the camera coupling rate, which collectively organize Bergsonian time structure. Furthermore, in this minimal set up, we studied how visual phenomena can manifest through feedback loops mediated by accumulated memories.

There exists a large corpus of video feedback related work, both artistic and scientific, beginning with Abraham (1976), to the well known work of Crutchfield (1984), to more recent work in Pixellated Video Feedback by Leach et al. (2003). As a starting point, perhaps the most complete list of video feedback projects is given at <http://www.videofeedback.dk/World/>.

In the following we introduce how we couple a chaotic neural network with a visual feedback system.

Visual feedback module: Instead of using physical video feedback which we did in MTM exhibition, a synthetic model simulates an internal mental feedback process, generating a new image frame, $I(x)$, as: $I_{n+1}(x) = LI_n(x) + L' \langle I_n(x) \rangle_x + sfI_n(bRx)$, where L is the intensity dissipation, L' is the spatial diffusion contribution, b is the zoom, R is 2D rotation, f is the aperture setting, and s is a luminance inversion parameter (Crutchfield, 1984).

Chaotic neural network architecture: The network design was previously explored in the MTM study. Here, images from the camera and the internal feedback are subsampled and imprinted into the first network layer weights, w_{ij} , of 400 neurons using a Hopfield approach (Hopfield, 1982): $\Delta w_{ij} = \rho \sum_{s=1}^M (2V_i^s - 1)(2V_j^s - 1)$, where V is a normalized pixel value, ρ is an input scale factor, and M is the number of images to imprint. A weight at step n is then: $w_{ij}^{n+1} = \psi w_{ij}^n + \Delta w_{ij}^n$, where $\psi \in [0, 1]$ is a forgetting parameter.

Outputs from the first layer are projected into the second layer of 9 neurons that are then mapped to the 8 video feedback parameters and coupled memory update and camera capture rate (as a function of the rate of change of the neuron output). The second layer uses a modified Hebbian learning rule; the change in weight w_{kj} by neuron p_j ; $k, j = 1, \dots, 9$ is: $\frac{dw_{kj}}{dt} = \gamma(\frac{dp_k}{dt} p_j - \alpha w_{kj} p_j^2)$, where $\frac{dp_k}{dt} = p_k^n - p_k^{n-1}$ emphasizes network structure change when input is most dynamic and α and γ are scale factors. Hebbian learning is performed on the weights between layers and within the second layer.

The update rule for all neurons in the network is: $p_k^{n+1} = r_1 p_k^n + (1 - \frac{1}{1 + e^{(q_k^n - p_k^n/\beta)}})$. Here $q_k^n = r_2 \sum_{j \neq k} w_{kj} p_j^n$, and r_1, r_2, β shaping the transfer function's behavior (Nozawa, 1992).

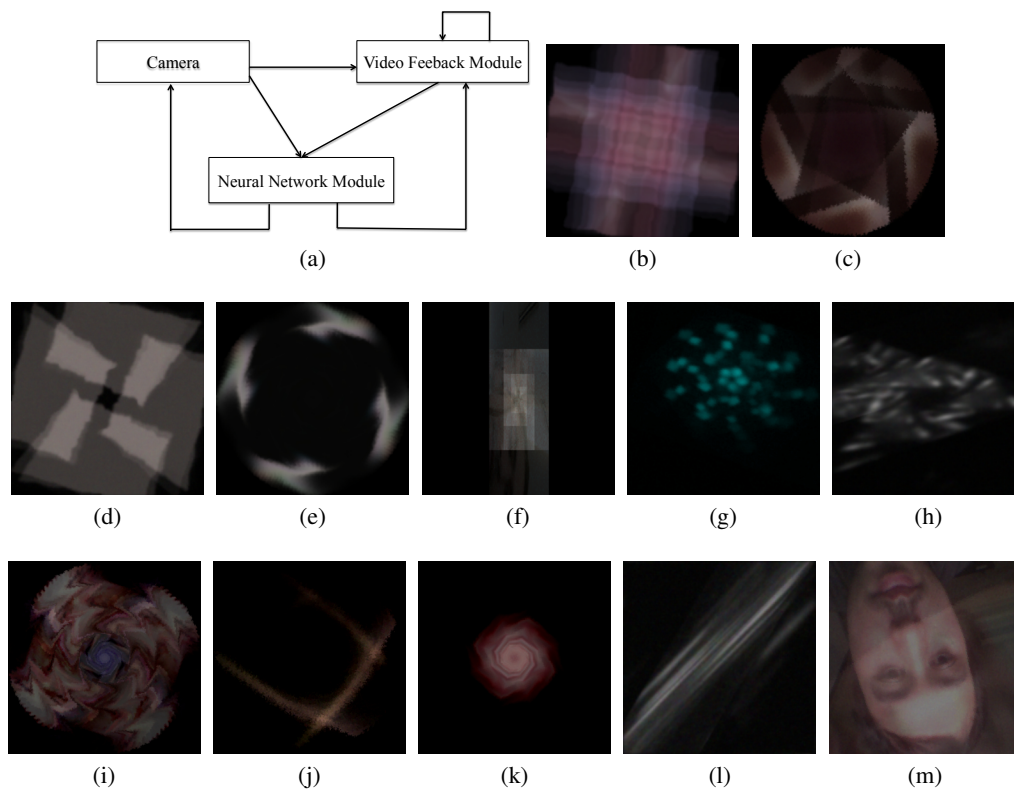


Figure 1: Screen captures of the system showing a variety of video-feedback phenomena, as well as a diagram showing overall system structure in 1a - arrows denote data flow.

Results: A selection of screenshots in Fig. 1 show a number of interesting video feedback effects such as phyllotactic and fractal like spirals, to spatial magnification and contraction in different dimensions. What cannot be captured from the static images is the interesting temporal evolutions of these patterns, both for a constant parameter set and for the transition between parameters.

As part of future work, the ability to access and record the system dynamics on the computer provides an opportunity to analyze the system from a scientific perspective. Additionally, could the type of visual feedback process explored in this work be a mechanism for *Entoptic* phenomena such as *Form Constants*? These are geometric patterns that people have described as being seen ‘within the eye’ during altered states of consciousness or hallucinations (Klüver, 1966).

References

- Abraham, R. (1976). Simulation of cascades by video feedback. In Hilton, P., editor, *Structural Stability, the Theory of Catastrophes, and Applications in the Sciences*, volume 525 of *Lecture Notes in Mathematics*, pages 10–14. Springer Berlin / Heidelberg.
- Crutchfield, J. P. (1984). Space-time dynamics in video feedback. *Physica D: Nonlinear Phenomena*, 10(1-2):229–245.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558.
- Ikegami, T. (2010). Studying a self-sustainable system by making a mind time machine. In *Workshop on Self-Sustaining Systems*, S3 ’10, pages 1–8, New York, NY, USA. ACM.
- Klüver, H. (1966). *Mescal and mechanisms of hallucinations*. University of Chicago Press, Chicago.
- Leach, J., Padgett, M., and Courtial, J. (2003). Fractals in pixellated video feedback. *Contemporary Physics*, 44(2):137–143.
- Nozawa, H. (1992). A neural network model as a globally coupled map and applications based on chaos. *Chaos*, 2(3):377–386.

Author Index

- | | | | |
|---------------------------------|---------------|-----------------------------|---------------|
| Adami, Christoph | 567, 581 | Castro, Leandro N. de | 509, 563 |
| Albertsen, Anders Nikolaj | 603 | Channon, Alastair | 317 |
| Alicea, Bradly | 147 | Chao, Lin | 519 |
| Allen, Peter B. | 441, 605 | Chatman, Jamie | 186 |
| Almeida, Diego J. D. | 509 | Chaumont, Nicolas | 581 |
| Anderson, Carlos J.R. | 3 | Chen, Xi | 441, 605 |
| Apolinário Jr., Antônio L. | 563 | Cheng, Betty H.C. | 171 |
| Arita, Takaya | 226, 301, 423 | Chirieleison, Steven | 605 |
| Armstrong, Cristobal | 431 | Chou, Andrew | 415 |
| Arnold, Solvi | 301 | Christensen, Anders L. | 251, 275, 553 |
| Asao, Akihiro | 523 | Church, George | 621 |
| Aubert, Nathanael | 544 | Cillari, Sonia | 637 |
| Auerbach, Joshua E. | 309, 629 | Cisneros, Luis | 283 |
| Barrick, Jeffrey E. | 473, 611 | Clark, Anne E. | 534 |
| Bartlett, Francis | 511 | Clark, Anthony J. | 325 |
| Bentley, Peter J. | 202 | Clune, Jeff | 387, 593, 631 |
| Berger, William | 587 | Conibear, Tim C. R. | 569 |
| Beslon, Guillaume | 536 | Cooper, Vaughn S. | 521 |
| Birattari, M. | 551 | Covert III, Arthur W. | 27, 32 |
| Blount, Zachary D. | 513 | Crailsheim, Karl | 597 |
| Blum, Jeremy | 387 | Cussat-Blanc, Sylvain | 491, 607 |
| Boenzli, Eva | 615 | D'Ambrosio, David B. | 379 |
| Boncella, James M. | 603, 617 | Davies, Paul C.W. | 283 |
| Bongard, Josh C. | 309 | Deboever, Chris M. | 534 |
| Borg, James M. | 317 | Deiters, Alex | 605 |
| Bramson, Aaron | 587 | Derryberry, Dakota Z. | 27, 32 |
| Bryson, David M. | 19, 585 | Dijk, Sander G. van | 333 |
| Bugallo, Monica Bello | 637 | Doan, Eugene | 631 |
| Burgos, Andres C. | 515 | Doncieux, Stéphane | 407 |
| Burns, Mark | 621 | Dorigo, M. | 551 |
| Bush, Eliot C. | 534 | Doursat, Rene | 349 |
| Canino-Koning, Rosangela | 517 | Downing, Keith L. | 37 |
| Cape, Jonathan L. | 603, 617 | Droop, Alastair | 45 |
| Carlson-Stevermer, Jared | 32 | Ducharme, Vincent | 449 |
| Carneiro, Jorge | 275 | Ducommun, Bernard | 465 |
| | | Duthen, Yves | 465, 607 |

- Dyer, Fred C. 178, 511, 567, 585
Dörr, Mark 583, 617
- Edson, Joseph B. 617
Egli, Richard 449
El-Hani, Charbel N. 509, 563
Elena, Santiago F. 111
Ellefson, Jared W. 609
Ellington, Andrew D. 441, 528, 605
.....609, 611
Ellis, Crystal N. 521
Enyeart, Peter J. 611
Eskridge, Brent E. 155, 163
- Fellermann, Harold 583, 603, 613, 617
Ferrante, Eliseo 551
Fisher, Steven 186, 587, 633
Flocken, Carissa 587
Floreano, Dario 530, 561
Flores, Cesar 532
Frenoy, Antoine 53, 218
Froese, Tom 457
Fryer, Craig S. 186
Fuchslin, Rudolf 613
- Garza, Mary A. 186
Gaucher, Eric A. 11
Gil, Benny 613
Goings, Sherri 171
Gold, Eric 387
Goldsby, Heather J. 171, 178
Gomes, Jorge 553
Goudarzi, Alireza 259
Grabowski, Laura M. 585
Gras, Robin 105
Grim, Patrick 186, 587, 633
Grouchy, Paul 59
Gudwin, Ricardo 589
- Hadorn, Maik 613, 615
Hamann, Heiko 597
- Hanczyc, Martin M. 613, 615
Harrington, Kyle I. 194, 555
Haynes, Lillian E. 534
Heywood, Malcolm I. 559
Hickinbotham, Simon 45
Higgs, Paul G. 625
Hintze, Arend 567
Holekamp, Kay 243
Homaifar, Abdollah 546
Honda, Hidehito 401
Hosoda, Kazufumi 523, 619
Hotz, Peter Eggenberger 613, 615
Hougen, Dean 137
Hudson, Robert 591
Huepe, Cristian 551
Hunicke-Smith, Scott P. 611
Hwang, Tony W. 528
- Ichihashi, Norikazu 525
Ichinose, Genki 557
Ikegami, Takashi 234, 457, 542, 639
Ito, Takashi 423
Izidoro, Venyton N. L. 509
- Jain, Ashish 341
Joachimczak, Michal 67, 349
Johnson, Brian 621
Joshi, Nikhil J. 357
- Kaçar, Betül 11
Kayama, Yoshihiko 526
Kazuta, Yasuaki 525
Keller, Laurent 530, 561
Kelly, Stephen 559
Kerr, Benjamin 178
Kiedrowski, Gunter von 613
Knibbe, Carole 536
Koch, Christof 357
Kowaliw, Taras 349
Kraaijeveld, Alexander R. 569
Kramash-Stettiner, Orly 635

- Krawec, Walter O. 364
 Kuehn, Timothy 372
 Kunstmann-Olsen, Casper 613

 Lai, Lawrence 621
 Lancet, Doren 613
 Laursen, Lone 583
 Ledbetter, Michael P. 528
 Legault, Claude Y. 449
 Lehman, Joel 75, 379
 Lenski, Richard E. 499, 513, 532, 538
 Li, Shuguang 431
 Lichocki, Pawel 530, 561
 Lichodziejewski, Peter 559
 Lim, Soo Ling 202
 Lima, Pedro U. 275
 Lin, Xiaoxia Nina 621
 Lipson, Hod 59, 387, 631
 Lobaskin, Vladimir 574
 Lobjois, Valérie 465
 Lohmann, Sara 387
 Loula, Angelo C. 509, 563, 589
 Luan, Hao 565
 Luga, Herve 465, 607
 Löffler, Philipp M. G. 617

 Ma, Singer 534
 Manahov, Viktor 591
 Manicka, Santosh 393
 Manser, Marta B. 571
 Marcotte, Edward M. 611
 Mariategui, Jose Carlos 637
 Marshall, James 573
 Matsuka, Toshihiko 401
 Matsuura, Tomoaki 619
 Maurer, Sara E. 603
 McCarthy, Corrine 267
 McCaskill, John 613
 McDermott, Matt 534
 McIver, Andrew 605
 McKay, Bob 91

 McKinley, Philip K. 325
 McMullin, Barry 83
 Meyer, Adam J. 609
 Meyer, Austin G. 473
 Meyer, Justin R. 532
 Michael, Loizos 210
 Miikkulainen, Risto 243, 341
 Miller, Jennifer M. 565
 Minty, Jeremy 621
 Misevic, Dusan 53, 218
 Monnard, Pierre-Alain 603, 613, 617
 Moore, Jared M. 325
 Mouret, Jean-Baptiste 593

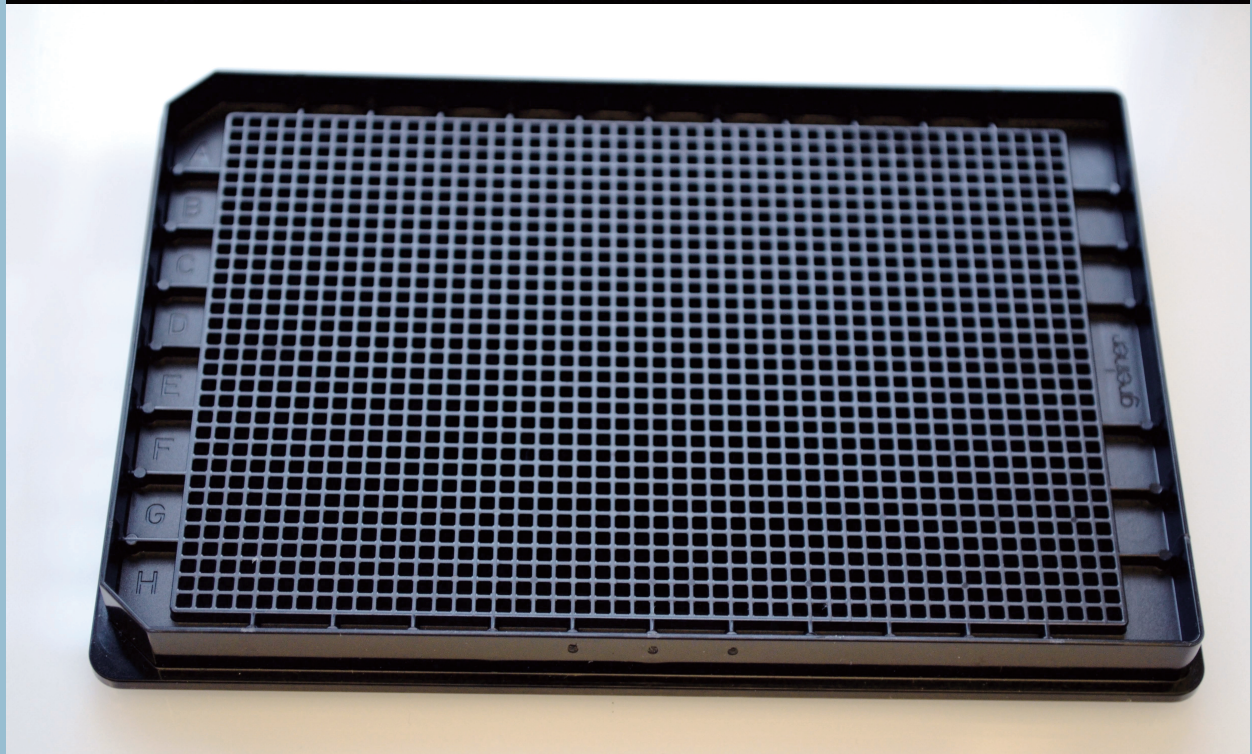
 Nishimoto, Keita 226
 Nitschke, Geoff 542
 Norman, Sally Jane 637
 Novak, Adam M. 534

 Ofria, Charles 19, 116, 171, 178
 499, 517, 585
 Ogai, Yuta 639
 Oka, Mizuki 234
 Oliveira, Sancho 251
 Ollion, Charles 407
 Olsen, Megan M. 555
 Olson, Randal S. 567
 Ozisik, A. Pinar 194

 Palmer, Michael E. 415
 Park, Jihyang 621
 Parsons, David P. 218, 536
 Pascalie, Jonathan 465, 607
 Pedersen, Pernille L. 617
 Peng, Annie Y. 519
 Penn, Alexandra 569
 Pennock, Robert T. 511, 585
 Perony, Nicolas 571
 Pilat, Marcin L. 423
 Pinville, Tony 407
 Polani, Daniel 333, 515

- Pollack, Jordan B. 194, 491
Punithan, Dharani 91
- Queiroz, Joao 589
- Rajagopalan, Padmini 243
Ramírez, Juan Camilo 573
Rang, Camilla U. 519
Rasmussen, Steen 583, 603, 613, 615, 617
Ratcliff, William 99
Rawal, Aditya 243
Reade, Christopher 186, 633
Reba, Aaron 473
Rebolleda-Gomez, Maria 99
Reggia, James A. 291
Ribeck, Noah 538
Richardson, Thomas O. 571
Rieffel, John 372
Risi, Sebastian 379
Rocha, Pedro L. B. da 563
Romensky, Maksym 574
Rossi, Louis F. 565
- Saito, Masaya 557
Sauro, Herbert M. 481
Sayama, Hiroki 576
Schlupp, Ingo 137
Schmickl, Thomas 597
Schmidt-Petersen, Sif 583
Schweitzer, Frank 571
Scott, Ryan 105
Serra, Neem 178
Shen, Chien-Chung 565
Shimohara, Katsunori 226
Siegelmann, Hava T. 555
Silva, Fernando 251
Simpson, Zack B. 441
Singer, Daniel J. 186, 587, 633
Sleight, Sean C. 481
Smith, Lane 27
Snyder, David 259
- Solé, Ricard V. 111
Song, Zhao 595
Stanley, Kenneth O. 75, 379
Staples, Rachel K. 521
Stovall, Gwendolyn M. 528
Stradner, Jürgen 597
Subramoney, Anand 341
Suzuki, Hiroaki 619, 623
Suzuki, Reiji 226, 301, 423
Suzuki, Shingo 523
Suzuki, Shinsuke 557
Svaneborg, Carsten 613, 617
Swarup, Samarth 267
- Taddei, François 53, 218
Tan, Xiaobo 325
Tanev, Ivan 226
Tarapore, Danesh 275, 561
Tavares, Emanuel M. C. 509
Taylor, Tim 540
Tenhaaf, Nell 637
Teuscher, Christof 259
Thomas, Stephen B. 186
Tononi, Giulio 357
Tournois, Sylvain 607
Travisano, Michael 99
Tsuda, Soichiro 623
Turgut, Ali Emre 551
- Urbano, Paulo 251, 553
Usui, Kimihito 525
- Valverde, Sergi 111
Vanouse, Paul 637
Vaughan, Richard T. 595
Virgo, Nathaniel 457
- Walker, Bess L. 116
Walker, Sara Imari 283
Wamberg, Michael 617
Wang, Harris 621

Wang, Jianxun	325	Workineh, Abraham	546
Watson, Richard A.	121, 569	Wrobel, Borys	67, 349, 599
Webb, Jeremy S.	569	Wu, Meng	625
Weitz, Joshua S.	532		
Wenseleers, T.	551	Yiannakides, Anastasios	210
Wentworth, John S.	534	Yomo, Tetsuya	523, 525, 619, 623
Wieczorek, Rafal	617	Yosinski, Jason	387, 631
Wilke, Claus O.	27, 32		
Williams, Hywel T.P.	129	Zagal, Juan Cristobal	431
Winder, Ransom K.	291	Zahadat, Payam	597
Witkowski, Olaf	542, 544	Zaman, Luis	499
Woehrer, Mark	137	Zaroff III, Ted	621
Woodward, Alexander	639	Ziock, Hans-Joachim	603, 617



Artificial Life 13 was hosted by the BEACON Center for the Study of Evolution in Action, at Michigan State University, East Lansing, Michigan, USA (<http://beacon-center.org>).

Cover design by Chris Corneal and David M. Bryson
 ALife 13 logo design by David M. Bryson and Charles Ofria
 Back cover photo by Sean C. Sleight (see paper on pg. 481)

978-0-262-31050-5
 Artificial Life 13, MIT Press, Cambridge 2012

ALIFE13
 evolution in action